



**LAUREA**  
AMMATTIKORKEAKOULU

*Uuden edellä*

# Verkkosovelluksen kehittäminen Profiili Palvelut Oy:lle asiakaskyselyiden tekemistä varten

Rönn, Toni

2014 Laurea Kerava



Laurea-ammattikorkeakoulu  
Kerava

Verkkosovelluksen kehittäminen Profiili Palvelut Oy:lle asiakas-  
kyselyiden tekemistä varten

Toni Rönn  
Tietojenkäsittelyn koulutusohjelma  
Opinnäytetyö  
Joulukuu, 2014

Rönn, Toni

### Verkkosovelluksen kehittäminen asiakaskyselyiden tekemiselle

Vuosi 2014

Sivumäärä

53

---

Tässä opinnäytetyössä tutkittiin verkkosovelluksen kehittämistä Profiili Palvelut Oy:lle asiakaskyselyiden tekemistä varten. Ennen tutkimusta suoritettiin tiedonkerääminen ohjelmoitavaa verkkosovellusta varten. Valmiissa sovelluksessa on valittu kuusi erilaista kysymystä, joiden vastaukset kerätään SQLITE3-tietokantaan, josta tieto viedään Excel-tiedostoon. Verkkosovellus oli ohjelmoitu laskemaan tulokset automaattisesti, jotta säästyttiin inhimillisiltä virheiltiltä ja vääränlaiselta palvelukehitykseltä.

Tutkimuksesta on rajattu pois kaikki muut tietotekniset ratkaisut paitsi SQLITE3-tietokanta, APAHCE-verkkopalvelin ja Hypertext Preprocessor (PHP)-ohjelmointi, sekä CSS-tyylitiedostot. Tutkimus on mahdollista toteuttaa eri tietotekniikka ratkaisulla, mutta tutkimukseen valitut tekniikat soveltuvat parhaiten tämän tutkimuksen tekemiseen.

Teoriaosuudessa on kerrottu Cascading Style Sheets (CSS)-tyylitiedostoista, SQLITE3-tietokannasta, APAHCE-verkkopalvelimesta ja PHP-ohjelmointikielen käytöstä. Lisäksi oli kerrottu asennus ja käyttöönotto näistä tekniikoista.

Opinnäytetyön tekemisessä on käytetty laadullista tutkimusta, jossa käytin toimintatutkimus menetelmää. Tutkinnan kohteena oli kehittää verkkosovellus asiakaskyselyiden tekemiselle monivalintakysymysten avulla. Vastaukset muokattiin numeraalisiksi. Toisena tehtävänä oli saada avoimella kysymyksellä sanallista tietoa palvelun kehittämisestä. Vastauksilla mitattiin prosentuaalisia tuloksia sekä sanallisten vastauksien tuloksia ristiin keskenään. Tutkimuksen tuloksista oli nähty se, että mitä palvelun osa-alueita on painotettu.

Rönn, Toni

**The business development with a network based application**

Year	2014	Pages	53
------	------	-------	----

---

In this bachelor's thesis was researched a network software's development with customer survey to Profiili Palvelut Oy. Before a research data have been collected to programmable network software. In the final program interface six different questions have been chosen. An application data are collected to SQLITE3 database. From database all the information is taken to an Excel file. The network application was programmed so that the results are calculated to be automatically and a human error and wrong type services development will be saved.

Everything else of information technology have been marked off in this research an exception SQLITE3 database, APACHE server and PHP programming language and CSS. The research is a possibility implement with in various information technology solutions and in research have been chosen techniques be suitable for the best in this research.

The Cascading Style Sheets (CSS), SQLITE3 database, APACHE server and PHP programming language were told in theoretical section. Also there was an installation and an introduction of these techniques.

In this bachelor's thesis qualitative research have been used in this research where action research methods have been used. In subject of investigation were developed a network application with multiple customer choice questions. The first five answers had been changed to numerical value. The second task was kept with an open question from verbal information to be able service quality develop. All the answers were measured in percentage terms and a verbal information results in every direction together. The result of the research had been seen that what parts of services in the result are emphasized.

Keywords: Apache, PHP, HTML, CSS, SQLITE3

## Sisällys

1	Johdanto.....	7
2	Keskeiset käsitteet ja lyhenteet .....	8
3	Työn tausta.....	8
3.1	Tavoite .....	8
3.1.1	Tutkimuskysymykset .....	8
3.1.2	Aihealueen rajausta .....	9
3.2	Palvelun laatu.....	9
3.2.1	Asiakaan palvelukokemus .....	9
3.2.2	Palautteen antamisen tärkeys .....	10
4	Teoriaa käytetyistä sovelluksista.....	10
4.1	Apache-palvelin .....	10
4.1.1	Apache-palvelimen tärkeimmät asetukset.....	11
4.1.2	Apachen virtuaalipalvelin.....	11
4.2	PHP-ohjelmointi .....	11
4.2.1	PHP-funktiot.....	12
4.2.2	PHP-komentotulkki.....	12
4.3	SQLITE3-tietokanta .....	12
4.3.1	SQLITE3 käytettävyys .....	12
4.3.2	SQLITE3-tietokannan helppokäyttöisyys.....	13
4.4	CSS-tyylitiedosto .....	13
4.4.1	CSS-asemointi .....	13
4.4.2	CSS-ulkoasu.....	13
5	Tutkimusmenetelmät.....	14
5.1	Kvantitatiivinen ja kvalitatiivinen tutkimus .....	15
5.1.1	Tutkimuksen reliaabelius .....	16
5.1.2	Tutkimuksen validius.....	16
5.1.3	Aineistonkeruu menetelmät .....	17
6	Toimintaympäristön käyttöönotto .....	17
6.1	Apache-palvelimen asentaminen ja käyttöönotto.....	17
6.2	Apache-palvelimen testaaminen .....	20
6.3	PHP:n asentaminen ja käyttöönotto .....	20
6.4	SQLITE3:n asentaminen ja käyttöönotto .....	21
6.5	PHP:n ja SQLITE3:n tarkistus.....	23
6.6	Verkkosovelluksen sivujen rakenne CSS tyylitiedostoilla.....	23
6.7	Tietokannan ja käyttöliittymän käyttöönotto.....	25
6.7.1	Kysymysten ja vastauksien lisääminen tietokantaan.....	26
6.7.2	Verkkosovelluksen ilmoitukset ja tietokanta .....	32

7	Verkkosovelluksen käyttö tablet laitteilla.....	42
8	Verkkosovelluksen vastauksien vieminen Excel-tiedostoon .....	44
9	Sivuille kirjautumisen hallinta .....	48
	9.1 Salasanan vaihtaminen hallintasivustolla .....	50
	9.2 Tietokantojen poisto .....	52
10	Yhteenveto ja johtopäätökset .....	53
	Lähteet .....	54
	Kuvat .....	55
	Kaaviot .....	77
	Liitteet .....	78

## 1 Johdanto

Tässä opinnäytetyössä tutkittiin verkkosovelluksen kehittämistä asiakaskyselyiden tekemistä varten. Ennen varsinaista tutkimusta suoritettiin tiedonkerääminen ohjelmoitavaa verkkosovellusta varten. Tarvittavien tietojen jälkeen valmiissa sovelluksessa on valittavana kuusi erilaista kysymystä joihin vastaaminen tapahtuu tablet-laitteilla. Tiedot kerätään tietokantaan, josta ne viedään Excel-tiedostoon. Verkkosovellus on ohjelmoitu laskemaan tulokset automaattisesti, jotta säästytään inhimillisiltä virheiltä ja vääränlaiselta palvelukehitykseltä.

Tutkimuksesta on rajattu pois kaikki muut tietotekniset ratkaisut paitsi SQLITE3-tietokanta, APAHCE-verkkopalvelin ja Hypertext Preprocessor (PHP), sekä Cascading Style Sheets (CSS)-tyylitiedosto. Tutkimus olisi ollut mahdollista toteuttaa eri tietotekniikka ratkaisulla, mutta tutkimukseen valitut tekniikat sopivat yksilöllisesti parhaiten tutkimuksen tekemiseen.

Teoria osuudessa käydään läpi CSS-tyylitiedostojen merkitystä verkkosovelluksen käyttöliittymän suunnittelussa. PHP-ohjelmointikielestä käydään läpi SQLITE3-tietokantahakuja, sekä APAHCE-verkkopalvelimen kanssa. Kaikkien edellä mainittujen tekniikoiden asennus ja käyttöönotto tulee käsiteltyä. Lisäksi mietitään käyttöliittymäasioita ja pohditaan käytettävyyttä.

Opinnäytetyön tutkimus oli kvalitatiivinen eli laadullinen tutkimus, jossa menetelmänä käytetään toimintatutkimusta. Tutkinnan kohteena oli selvittää verkkosovelluksen kehittämistä asiakaskyselyiden tekemiselle asiakkailta kerättyjen monivalintakysymysten avulla, joiden vastaukset muutetaan numeraalisiksi. Toisena tehtävänä oli saada yhdellä avoimella kysymyksellä sanallista tietoa palvelun kehittämisestä. Vastauksilla mitataan prosentuaalisia tuloksia sekä sanallisten vastauksien tuloksia ristiin keskenään.

## 2 Keskeiset käsitteet ja lyhenteet

(X)HTML	Hypertext Markup Language. Dokumenttien merkintäkieli
CSS	Cascading Style Sheets. Mahdollisuus hallinnoida sivujen ulkoasua
HTTP	Hypertext Transfer/Transport Protocol. Hypertekstin
Apache	Apache HTTP Server. HTTP-palvelinohjelma
SQLITE3	Kevyt tietokantasovellus, jossa ei tarvita erillistä tietokantapalvelinohjelmistoa
NOTEPAD++	NOTEPAD++ on ilmainen GPL-lisenssiin pohjautuva teksti- ja lähdekoodieditori
GPL	General Public License on yleisin avoimen lähdekoodin ohjelmistoissa käytetty lisenssi
PHP	Hypertext Preprocessor

## 3 Työn tausta

Aihe tutkimuksen tekemiselle löytyi liiketalouden opintojen kautta. Tutkimuksessa halusin hyödyntää myös ohjelmointi kokemustani. Näiden kahden asian yhdistämisen, jonka avulla voidaan tehdä esimerkiksi palveluiden laatua mittaavia kyselyjä.

### 3.1 Tavoite

Tavoitteena oli verkkosovelluksen suunnitteleminen ja toteuttaminen asiakaskyselyiden tekemiselle. Tarkoituksena oli saada helposti muokattavissa oleva ohjelmisto, jota pystyy käyttämään ilman ohjelmointi kokemusta. Lisäksi ohjelmiston pitäisi toimia ilman Internet-yhteyttä paikallisella palvelinkoneella omassa verkossa, johon otetaan yhteys tietokoneella tai tablet-laitteella. Toiminnoiltaan verkkosovelluksen oli tarkoitus olla helppokäyttöinen. Ohjelmiston käyttöliittymän tulisi antaa asiakkaalle viesti siitä, että kysymysten vastaaminen ei kestä pitkään. Asiakkailta kerätty tieto pitäisi olla tallennettavissa Excel-tiedostoksi, josta sitä olisi helppo hallita ja tehdä päätöksiä näihin tietoihin perustuen.

#### 3.1.1 Tutkimuskysymykset

Opinnäytetyöni tutkimuskysymyksenä on, että kuinka verkkosovelluksen avulla voidaan helpottaa yrityksen palveluiden kehittämiseen liittyen kyselyiden tekemistä. Miten asiakaskyselyiden vastaukset olisivat helpoiten hyödynnettävissä.



### 3.1.2 Aihealueen raja

Tässä opinnäytetyössä käytetään SQLITE3-tietokanta, APAHCE-verkkopalvelin ja PHP-ohjelmointia, sekä CSS-tyylitiedostoja. Tutkimus olisi ollut mahdollista toteuttaa eri tietotekniikka ratkaisulla, mutta tutkimukseen valitut tekniikat sopivat parhaiten tämän tutkimuksen tekemiseen. Tarkoituksena on antaa erilainen toimintatapa hankkia tietoa asiakkaista ja palveluiden kehityksestä, eikä pitää sitä ainoana vaihtoehtona.

## 3.2 Palvelun laatu

Asiakkaat muodostavat palvelun laadusta aina jonkin kokonaiskuvan, jotka mainitaan esim. termeillä ”laadukas”, ”hyvä” tai ”tyytyttävä”. Lopputulos tutkimuksissa on yksinkertainen mitata ja se antaa kokonaiskuvan asiakkaiden tuloksista. Useasti on hyvä tarkastella asiakkaiden laatuarviota tarkemmin. Silloin tarkastellaan laadun eri osa-alueita. Tarkempi tarkastelu on hyvä tehdä kun halutaan saada selville miksi asiakkaiden kokonaisarvo on jollakin tietyllä tasolla. Asiakasvastauksien tarkastelu antaa vihjeitä siitä mitä palvelulaadun parantamiseksi voidaan tehdä. Palveluiden laatuun keskittyvien tutkimusten mukaan on olemassa viisi osa- aluetta, joita asiakkaat useasti käyttävät palvelun laatua arvioidessaan. 1) Varmuus: lupaus- ten pitäminen, odotusten vahvistuminen, hyvän toiminnan jatkuvuus. 2) Palveluvalmius: hen- kilöstön innokkuus palveluun, palvelun nopeus. 3) Luottamus: mahdollisuus saavuttaa luotta- muksellinen suhde asiakkaaseen. 4) Henkilökohtainen huomioiminen: asiakkaan kunnioitus, asiakkaan tarpeiden huomioiminen. 5) Fyysiset puitteet: Toimitilat, laitteistot, sisustus, hen- kilöstö ja kirjallinen materiaali. Palveluiden laatua arvioitaessa kaikki viisi mainittua osa- aluetta esiintyvät useimmissa palveluissa. Se kuinka asiakas kiinnittää huomionsa yrityksen palvelulaatuun vaihtelee. Toisinaan asiakkaat perustavat laatuarvionsa kaikkiin viiteen teki- jään. Eri tilanteissa asiakkaat voivat käyttää yhtä tekijää. (Lämsä & Uusitalo 2009, 60.)

### 3.2.1 Asiakaan palvelukokemus

Asiakas on tekemisissä isoon joukkoon yrityksen työntekijöistä ja saa hyvän tai huonon palve- lukokemuksen koko organisaation työntekijöiden perusteella. Yksi huonosti motivoitunut työn- tekijä pystyy vaikuttamaan huonolla käytöksellä asiakkaan hyvään elämykseen. Mahdollisuus siihen, että asiakas valitsee seuraavalla kerralla kilpailevan yrityksen, jossa palvelukokemus on positiivinen. Saadun palvelukokemuksen laatu on ratkaiseva tekijä sille palaako asiakas liikkeeseen takaisin vai asioiko asiakas kilpailevassa yrityksessä. Palvelukokemuksen laadun ratkaisee se, miten hyvin yrityksen työntekijät ovat motivoituneet ja sen ratkaisee heidän johtamisensa laatu. (Hiltunen 2012, 19.)

### 3.2.2 Palautteen antamisen tärkeys

Pulma tai ongelma on onnittelun aihe. Valitukset tuovat esille pulmatilanteet ja ongelmakohdat, joiden tiedostaminen on palveluiden kehittämisen kannalta olennaista. Jos ongelmat eivät tule esille, niin niitä ei voi myös korjata. Valitukset ovat yksi tärkeimmistä palveluiden kehittämistä olevista asioista. Valitusten käsitteleminen ja oikeanlainen tunnistaminen tuovat yritykselle mahdollisuuden työyhteisön oppimiseen. Avoin keskustelu virheistä on yritykselle parhaita palveluiden kehittämisen tapoja, jossa esimies on ratkaisevassa asemassa. Palvelulaadun kehittämisen osalta on tärkeää, että yrityksessä sisäistetään ajatus siitä kuinka virheistä voi oppia. Puolustuskannalla oleminen tai ongelmien piilottaminen eivät edistä palveluiden kehittämistä. Asiakkaalle valituksen tai palautteen tekeminen voidaan kokea vaikeaksi, jolloin valittaminen on tehtävä helpoksi. (Lämsä & Uusitalo 2009, 73.)

## 4 Teoriaa käytetyistä sovelluksista

On olemassa ohjelmistopaketteja jotka sisältävät Apache -verkkopalvelimen ja PHP-ohjelmointikielen tuen, sekä SQLITE3-tietokantaominaisuuden. Asennuspakettien mukana tulee paljon ylimääräisiä toimintoja kuten postipalvelin sekä muita ominaisuuksia. Yksin asennettuna Apache-verkkopalvelimesta saadaan turvallisempi ilman ylimääräisiä ohjelmistoja. CSS-tyylitiedostot toimivat suoraan HTML-tiedostojen kanssa ilman Apache-verkkopalvelinta.

### 4.1 Apache-palvelin

Apache-palvelin on avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma. Apache on myös monipuolinen Internetissä toimiva palvelinohjelmisto. Apachen Unix - ja Windows versiot poikkeavat hieman toisistaan Apachen Unix-versiot ovat toimivampia ja paremmin testattuja. Windows- ja Unix-käyttöjärjestelmillä toimivat versiot ovat melkein samanlaiset, mikä mahdollistaa samanlaisen tuotekehityksen molemmissa alustoissa. Palvelinohjelmat kuuntelevat käyttäjien lähettämiin pyyntöihin tietyissä porteissa. Kaikki portit ovat numeroitu ja varsin pitkälle myös standardoitu. Apache-palvelin vastaa selaimille portissa 80. Teoriassa yhdellä palvelimella on mahdollista olla käynnissä monia palvelinohjelmia, koska käytettävissä on kymmeniä tuhansia portteja. Vaikka käytössä olisi tehokas tietokone ja paljon muistia, niin silti resurssit loppuvat paljon ennen kuin vapaana olevat portit. Virallisen termin mukaan palvelinohjelma käsittelee tietopyyntöjä tietyssä portissa esimerkiksi portissa 80. Portti ei kuitenkaan ole mikään fyysinen laite tai liitäntä, vaan se on ainoastaan ohjelmistovääritystä. Porttikohtaisesti palvelimella voi olla useita erilaisia www-palvelimia, joista jokainen porttimääritys voivat johtaa täysin erilaiseen www-palveluun. (Heinisuo 2004, 14-20.)

#### 4.1.1 Apache-palvelimen tärkeimmät asetukset

Apachen asetukset asetetaan httpd.conf-tiedostossa. Palvelimen toimintaa ohjaavia määrittäjiä on satoja. Kaikkia asetuksia ei tarvitse Apachen-palvelimen peruskäytössä käyttää ja tarvittavat muutokset tehdään muokkaamalla https.conf-tiedostoa. Tiedosto on muokattavissa tekstieditorilla ja muutoksia kannattaa tehdä ainoastaan jos oikeasti tietää mitä tekee. Servername-määrittäksellä kerrotaan Apache-palvelimelle palvelimen domainnimi. Serverroot-määrittäksellä asetetaan Apache-oletushakemisto. Servertokens-määrittäys antaa mahdollisuuden teknisten ja versiotietojen julkaisemisen server-otsikkotiedossa. Timeout-määrittäys on arvo joka määritetään ajan sekunneissa, jonka jälkeen yhteys selaimen katkaistaan. Selainten tuottamat GET- ja POST-pyynnöt toimivat sekunneissa tai sekunnin murto-osissa. Määrittäys-Options, AllowOverride ja Directoryn avulla kerrotaan, mitkä toiminnot ovat sallittuja ja mitkä ei. (Heinisuo 2004, 208-209.)

#### 4.1.2 Apachen virtuaalipalvelin

Virtuaalipalvelin tarkoittaa sitä, että yhdellä tietokoneella Apache-ohjelma voi hallinnoida useita erinimisiä www-palvelun osia. Palvelimella on mahdollista olla useita domainnimiä, jotka voivat hallinnoida eri organisaatioiden www-palveluja. Internetissä palvelimella on mahdollista olla useita IP-osoitteita. Kaikilla osoitteilla voi olla nimipalvelussa useita nimiä, jotka ohjaavat eri IP-osoitteisiin. Apache-palvelimella on mahdollista määrittellä IP-osoitteisiin perustuvia virtuaalipalvelimia (IP-based Virtual Hosts), sekä yhteen IP-osoitteeseen määritettyjä virtuaalipalvelimia. Tällöin kaikkien virtuaalipalvelimien Internet-sivut sijaitsevat omissa hakemistoissa palvelimen kovalevyllä. (Heinisuo 2004, 213-214.)

#### 4.2 PHP-ohjelmointi

Hypertext Preprocessor (PHP)- kielen käskyjä voi kirjoittaa HTML-koodin sisälle. PHP-ohjelmoinnin komentoja on mahdollisuus kirjoittaa mihin tahansa HTML-sivustolla. Lisäksi PHP-kielillä on mahdollista kirjoittaa tai lukea palvelimella olevia tekstimuotoisia tiedostoja. PHP on tulkittava kieli eli Internet sivujen sisällä oleva PHP-koodi luetaan aina palvelimella kun WWW-palvelin lähettää sivun tiedot selaimelle. PHP sisältää kaikki ohjelmointikielille tutu komennot, sekä tyypilliset rakenteet. Komennot voivat olla esimerkiksi for- ja while-silmukoita tai pitää sisällään if-lauseita, sekä muuttujat ja funktiot. PHP-ohjelmointi voidaan toteuttaa myös oliopohjaisesti, eli sovellukset voidaan ohjelmoida luokkina perinteisen funktioihin perustuvan ohjelmoinnin sijaan. (Heinisuo 2004, 16-17.)

#### 4.2.1 PHP-funktiot

Funktio on ohjelmointikielen rakenne mikä antaa mahdollisuuden toistaa rutiinien kokoamisen yhdessä ohjelmistokoodin kohdassa. Funktio tehdään kerran ja kaikille funktioille annetaan omat nimet, joita on mahdollista kutsua aina, kun sitä tarvitaan. Funktiolla on aina nimi ja sille voidaan määritellä omia parametreja sekä palautusarvo. On siis mahdollista kirjoittaa Print-komento tai PrInt- komento isoilla ja pienillä kirjaimilla. Myöskään funktiot eivät välitä kirjaimien koosta. Muuttujien kohdalla asia on toinen. Esimerkiksi \$nimi-muuttujaa ei voi kirjoittaa mitenkään muuten tai se on kokonaan eri muuttuja. (Heinisuo 2004, 51-56.)

#### 4.2.2 PHP-komentotulkki

PHP:ssä oleva php.exe-tiedosto on mahdollista käynnistää MS-DOS-ikkunassa. PHP-kielillä kirjoitettuja sovelluksia voidaan ajaa myös ilman www-palvelinta. Komentorillä on käytössä kaikki samat PHP-komennot, tietokantayhteydet, sekä muut toiminnot kuin www-palvelimella. (Heinisuo 2004, 178.) Erilaisia toimintoja on myös mahdollista sisällyttää BAT-tiedostojen sisälle. BAT-tiedostoon voidaan tehdä ennalta samoja käskyjä kuin mitä suoritetaan komentoriviltä. BAT-tiedoston voi tehdä tekstieditorilla, mikä lopuksi nimetään esimerkiksi tiedosto.bat-nimiseksi.

### 4.3 SQLITE3-tietokanta

SQLITE3-tietokanta on yksi PHP5 uusista ominaisuuksista joita sillä on. SQLITE3-tietokanta on parhaimmillaan kevyiden tietokantasovellusten käyttöön soveltuva tietokanta. SQLITE-tietokanta eroaakin esim. MySQL-tietokannasta monella eri tavalla. SQLITE3-tietokanta ei tarvitse ylimääräisiä tietokantapalvelinohjelmistoasennuksia ja sen muistivaatimukset ovat pienet. SQLITE3-tietokanta näkyy tiedostona muiden tiedostojen joukossa ja niitä voidaan lukea ja kirjoittaa esim. PHP5-ohjelmointikielillä. SQLITE3-tietokanta on erinomainen valinta yksinkertaisien ohjelmistojen tietokannaksi. (Heinisuo 2004, 165.) SQLITE3-tietokannassa toimivat samat SQL-tietokannasta tutut käskyt.

#### 4.3.1 SQLITE3 käytettävyys

SQLITE3-tietokantaa on mahdollista käyttää myös tablet-laitteissa ja puhelimissa eri ohjelmointikielen tietokantana. SQLITE3-tietokantaa ei tarvitse asentaa, niin kuin monia muita tietokantoja. SQLITE3 toimii monissa eri käyttöjärjestelmissä kuten, Linux, Windows, Mac OS X, sekä muissa järjestelmissä. SQLITE3-tietokanta sopii hyvin pienille ja keskikokoisille WEB-sivuille tietokannaksi. SQLITE3:n mukana tulee myös command-line interface (CLI)- komentokehoteohjelma millä voi hallinnoida SQLITE3-tietokantaa.

#### 4.3.2 SQLITE3-tietokannan helppokäyttöisyys

SQLITE3 ei tarvitse erillistä asennusta ennen kuin sitä voidaan käyttää. Tietokoneen käynnistyksen jälkeen SQLITE3 ei vaadi Windows-palveluista erillistä palveluiden pysäyttämistä tai käynnistämistä, eikä asetusten määrittämiä. Helppokäyttöisyys näkyy hyvin myös siinä, että SQLITE3:n käytössä ei tarvita erillistä järjestelmänvalvojaa tekemään oikeuksille olevia määrittämiä, jotta tietokantaa voitaisiin käyttää. Myöskään mitään palautustoimintoja ei tarvitse tehdä jos tietokone sammuu kesken kaiken tai järjestelmä kaatuu jostain syystä.

#### 4.4 CSS-tyylitiedosto

Cascading Style Sheets (CSS) on tyylitiedosto, jolla on mahdollisuus hallinnoida helposti Internet-sivustojen ulkoasua ja pitää erillään sivuston ulkoasu muusta sisällöstä. Ulkoasuelementit esim. font-elementti on mahdollista poistaa ja ulkoasua voidaan hallinnoida CSS:n avulla taulukoiden sijaan. Kaikki merkinnät on mahdollista tehdä vieläkin yksinkertaisemmin. (Collison, Budd & Moll. 2006, 3.) CSS-tyyliin avulla voi kohdistaa eri tyyliä omiin elementteihin ja niissä voidaan käyttää ID- ja class-valitsimia. Nimien mukaisesti valitsijat kohdistuvat ID- tai class-nimellä elementteihin. ID-valitsimet asetetaan toimimaan risuaita merkillä ja class-valitsimet pisteellä. (Collison, Budd & Moll. 2006, 4-35.)

##### 4.4.1 CSS-asemointi

CSS:llä on kolme perusasemointisysteemiä, joihin kuuluu normaali, kelluva ja absoluuttinen asemointi, joilla voidaan hallita sivustolla olevien laatikoiden käyttäytymistä. CSS:llä pystyy hallitsemaan Internet-sivun ulkoasua, jossa tekstin asetukset ovat erillään varsinaisesta HTML-koodista. Semanttinen eli rakenteellinen merkkkaus mahdollistaa kehittäjille monia tärkeitä etuja. Rakenteellisia sivustoja on helpompi hallinnoida kuin ulkoasuun keskittyviä. Semanttinen merkintä mahdollistaa helpomman tavan ohjata elementtejä, joita halutaan muotoilla. Semanttisuus tuo rakennelleisuutta ja luo perimmäisen runkorakenteen, jonka varaan voidaan rakentaa. Elementtejä on mahdollista muotoilla suoraan ilman, että pitäisi luoda muita tunnistimia jolla välttyään tarpeeton koodin paisuminen. (Collison, Budd & Moll. 2006, 4-35.)

##### 4.4.2 CSS-ulkoasu

Mukautuvassa ulkoasussa Internet-sivusto ilmoitetaan prosentteina pikseleiden sijasta. Tämä mahdollistaa ulkoasun skaalauksen, eikä välitä näytön asetuksista. Kun selainikkunaa suurennetaan, myös sarakkeet levenevät. Muokkautuvilla ulkoasuilla on myös omat ongelmansa. Pieniksi tehdyt ikkunanleveydet pienentyvät todella kapeaksi ja tällöin sisällöstä on vaikea saada

selvää. Sivuston koko leveyden käyttämisen sijasta voidaan käyttää sivuston leveydessä määrittystä 85 prosenttia. Täytteen ja marginaalin käyttämisessä voidaan käyttää prosentteina. 85 prosentin käyttö toimii hyvin kaikilla selaimilla, joten sitä kannattaa käyttää. Joustavat ulkoasut toimivat siten, että elementin leveys määritetään suhteessa selaimen leveyden sijasta kirjasin kokoon. Leveyksien asettaminen em-yksikköinä jolloin kirjasinkoko kasvaa kun ulkoasua skaalataan. Tämä mahdollistaa rivipituuksien pysyvän sopivan mittaisina, joka sopii hyvin heikkonäköisille ihmisille. (Collison, Budd & Moll. 2006, 145-147.)

## 5 Tutkimusmenetelmät

Vilkan (2006, 70) mukaan toimintatutkijalla on yhtäaikaaisesti erilaisia rooleja tutkimuksen aikana. Tutkija työskentelee tutkijana ja toimii muutostavoitteisena aktivoijan roolissa. Tutkijana tutkija miettii kaikki tarvittavat työvaiheet, joihin tutkimuksen aikana tutkimusaineistoa kerätään. Tutkimuksessa käydään aktivoivan ja osallistavan työskentelyn aikana toiminnassa mukana olevaa havainnointia sekä kerätään erilaisia haastatteluja tutkittavasta aiheesta. Tutkijan on hyvä kerätä myös sellaista tietoa tutkimuksessa mukana olevasta organisaatiosta, mikä auttaa tutkimuksen onnistumista mahdollisimman hyvin. Aktivoivassa osallistuvassa roolissa kerätään eri työvaiheissa olevaa tietoa. Havaintoja tehdään tutkimuksen alaisista työvaiheista sekä kesken tutkimuksen että muuttuneesta tilanteesta. Tutkimuksen aikana tutkijan toiminta on muutostavoitteinen tutkittavaa asiaa ja siihen osallistuvia jäseniä kohtaan. Tutkimuksessa mukana olevat jäsenet yritetään saada muutostavoitteisiksi tutkijan rooliin. Tutkijan rooliin kuuluvia asioita ovat välineet ja kirjallinen materiaali tutkimuskohteesta sekä menetelmät että menetelmät. Lisäksi toimintatutkimuksessa on olemassa toisenlaisia rooleja, kuten tieteellinen asiantuntija, konsultti, aktivoiva ja osallistuva tutkija. Nykyisin toimintatutkijat toimivat enemmän tieteellisen asiantuntijan roolista osallistuvaan tutkijaan. Tieteellinen asiantuntija toiminnallaan ohjaa tutkittavan aiheen muutuskokeiluja ja soveltaa teoriaa ja käytäntöä, sekä testaa tekemiään muutoksia. Tutkija toimii kuitenkin ulkopuolisen silmin eri muutosvaiheiden aikana ja toimii arvioijana. Tutkija konsulttina toimii pääsääntöisesti auktoriteettina jonka tehtävänä on johtaa muutosta koskevia menetelmiä. Konsulttina hän kertoo miten toimitaan eri muutostilanteissa ja saadaan aikaan tuottavuuden tavoitteet.

Käyttäjien työn ja tehtävien kautta opitut asiat ja käytännöt ovat erilaisia kuin mitä suunnittelijat ovat oppineet. Tämän takia kaikkiin tuotekehitysprojekteihin sisältyy käyttäjien toiminnan seuraaminen, jotta tiedetään käyttäjien tekemät toiminnot kyseisen tuotteen kanssa ja mitkä voisivat olla varteen otettavia virhetilanteita. Käyttäjien toimintaa täytyy tutkia ennen ja jälkeen tuotteen suunnittelun sekä sen aikana. Ennen tuotekehitystä käyttäjien toimintaa pyritään arvioimaan havainnoimalla, lokitiedostoilla, haastattelemalla tai millä muulla keinolla tahansa. Ennen tuotteen suunnittelua pitäisi olla jonkin näköinen käsitys tulevista

käyttäjistä, jotka ryhmitellään käyttäjäryhmiin. (Sinkkonen, Kuoppala, Parkkinen & Vastamäki ym.2006, 29.)

Opinnäytetyön tutkimusmenetelmänä on kvalitatiivinen eli laadullinen tutkimus, jossa menetelmänä käytetään toimintatutkimusta. Tutkimuksessa tutkittiin verkkosovelluksen kehittämistä asiakaskyselyiden tekemistä varten. Tutkimuksessa verkkosovellukseen tarvittava ohjelmoitava tieto on kerätty puhelimitse ja sähköpostilla. Varsinaisen tutkimuksen tekemiseen tarvittiin verkkosovellus toimimaan halutulla tavalla. Sovellus kerää asiakkailta monivalintakysymysten vastaukset, jotka muutetaan numeraalisiksi. Toisena tehtävänä oli saada vain avoimella kysymyksellä sanallista tietoa palvelun kehittämisestä. Vastauksilla mitataan prosentuaalisia tuloksia sekä sanallisten vastauksien tuloksia ristiin keskenään palveluiden kehityksessä. Ohjelmiston yhtenä tärkeimpänä ominaisuutena on vastauksien automaattinen laskenta, jossa päästään aina luotettavaan tutkimustulokseen ilman, että tulokset olisi minkä tahansa syyn takia henkilöstön laskemana tehty virheellisesti. Ohjelmiston luotettavuus on testattu koekysymyksillä ja tulokset on laskettu sekä manuaalisesti, että ohjelmiston tekemänä. Mikäli kummatkin toimintatavat ovat päätyneet testin aikana samaan tulokseen, niin ohjelmiston toiminta on voitu todentaa toimivaksi.

## 5.1 Kvantitatiivinen ja kvalitatiivinen tutkimus

Kvantitatiivisia ja kvalitatiivisia tutkimusmenetelmiä on hankala erotella keskenään toisistaan. Molemmat menetelmät toimivat toisiaan tukevinä menetelminä, ja näin ollen niiden ei myöskään ole tarkoitus kilpailla paremmuudesta keskenään, koska kummallakin toimintatavalla voi tuoda oman panoksen tutkimuksiin mutta eri tavalla kerrottuna. Kvalitatiivista tutkimusta voidaan käyttää tietojen kartoittamiseen kvantitatiiviselle tutkimukselle. Tällä tavalla toimimalla voidaan varmistaa tutkimuksissa olevien osa-alueiden oikeellisuus tutkimuksen ongelmien kannalta. Lisäksi oikean toimintamenetelmän varmistaminen tutkimusta suorittavalle henkilölle on tärkeää. Kvantitatiivisen tutkimuksen toteuttaminen ennen kvalitatiivista menetelmää voi olla laaja kvantitatiivinen lomakepohjainen tutkimus millä mahdollistetaan tehdä miellyttäviä vertailevia ryhmiä kvalitatiivisia haastatteluja varten. Kvantitatiivisessa tutkimuksessa kerrotaan, että se pitää sisällään numeroista koostuvia tietoja ja kvalitatiivinen tutkimus sanallisia merkityksiä. Tavoitteena ei ole laittaa molempia menetelmiä toistensa vastakohtiksi vaan täydentää toisiaan. Numerot ja sanalliset merkitykset ovat toisistaan riippuvaisia, jolloin tulokset ovat kerrottu eri tavoilla. Numerot sisältävät merkityksistä sisältävän käsitteellistämisen ja sanalliset merkitykset sisältävät käsitteellisiä havaintoja voidaan ilmaista numeroin. Mittaaminen sisältää kaikilla tasoilla sekä kvalitatiiviset että kvantitatiiviset osa-alueet. (Hirsijärvi, Remes & Sajavaara. 2007, 133.)

### 5.1.1 Tutkimuksen reliaabelius

Tutkimuksessa pyritään minimoimaan väärien tuloksien syntymistä, jolloin tulosten luotettavuus ja pätevyys vaihtelevat. Tulosten eroavaisuuksien vuoksi kaikissa tutkimuksissa on päämääränä tehdyn tutkimuksen luotettavuus. Mikäli tutkimuksen lopputulos on toistettavissa, niin silloin voidaan käyttää merkitystä reliaabelius. Tutkimuksen reliaabelius tarkoittaa, että mahdollisuus saada sattumanvaraisia tuloksia ei toteudu. Reliaabelius voidaan varmistaa monella eri tavalla. Jos useampi arvioija toteaa samanlaisen tuloksen, niin tulos on silloin reliaabeli, tai tutkittavana olevaa henkilöä tai ilmiötä tutkitaan eri aikoina ja päädytään samaan tulokseen, voidaan silloin todeta tulokset reliaabeleiksi. Lisäksi kvantitatiivisessa tutkimuksessa on kehitelty toisistaan eroavia tilastollisia toimintatapoja, joiden avulla pystytään arvioimaan mittareiden luotettavuutta. Useilla tieteenaloilla on myös tuotettu maailmanlaajuisesti testattuja mittareita, joilla voidaan kohottaa mittauksen laatua ja joiden kanssa päästään luotettavasti tarkistamaan maailmalla saatuja testituloksia. (Hirsijärvi, Remes & Sajavaara. 2007, 226.). Verkkosovelluksessa on varmistettu reliaabeliuksien toteutuminen, jossa on käyty läpi kaikki työvaiheet joita verkkosovelluksen käyttö edellyttää. Kaikki verkkosovelluksen kysymykset ja vastaukset on kirjoitettu paperille ja katsottu, että paperilla olevat tiedot vastaavat tietokannassa olevia tietoja. Tämä prosessi on toteutettu ohjelmiston eri kehitysvaiheissa, kun muutos on ollut koodissa merkittävä.

### 5.1.2 Tutkimuksen validius

Tutkimuksissa pitää aina varmistaa validius eli pätevyys. Validius määrittää mittarin tai tutkimusmenetelmän mahdollisuutta toimia tiettyjen viitearvojen mukaan, jotka ovat sallittuja, jossa testattavana olevaa asia voi toimia. Mittarit ja toimintatavat eivät aina vastaa käytäntöä, jota tutkija luulee tutkivansa. Esimerkiksi asiakkaalle tarkoitettuihin kysymyksiin saadaan tulokset, vaikka vastaajat ovat voineet käsittää kysymykset tai mahdollisen testitilanteen eri tavalla kuin tutkija on kuvitellut. Mikäli tutkija tarkastelee tuotettuja tuloksia alkuperäisen ajattelumallinsa mukaisesti, ei vastauksia pystytä käsittelemään tosina ja pätevinä. Mittari voi tuottaa tuloksiin virhettä mikäli alussa, keskellä tai testin lopussa on tehty virheitä. (Hirsijärvi, Remes & Sajavaara. 2007, 226.). Validiuksen toteutuminen on varmistettu niin, että verkkosovellus on suorittanut tarvittavat matemaattiset laskut ja niitä on verrattu testihenkilön tekemiin laskuihin asiakasvastauksista. Molemmilla laskutavoilla on päästy samaan tulokseen joka kerta ja tämä työvaihe on toistettu ohjelmiston eri kehitysvaiheissa.



### 5.1.3 Aineistonkeruu menetelmät

Toimintatutkimuksessa käytetään eri menetelmiä ristiin ja eri tilanteissa hankittua tutkimustietoa. Tutkija havainnoi ja tekee niistä kenttämuistiniipanoja, sekä haastattelee tutkimuksessa mukana olleita henkilöitä, sekä kerää erilaisia kirjallisia dokumentteja. (Hirsijärvi, Remes & Sajavaara. 2007, 187.) Kyselyjen tekeminen tutkimuksen aineiston saamiseksi tunnetaan survey-tutkimuksen olennaisena menetelmänä. Survey-tutkimuksessa kohdehenkilöt muodostavat otoksen tai perusjoukon tutkimuksessa. Kerätty tutkimusaineisto survey-tutkimuksen avulla käsitellään yleensä kvantitatiivisesti. (Hirsijärvi, Remes & Sajavaara. 2007, 188.) Haastattelu toimii ainutlaatuisena tiedonkeruumenetelmänä, jolloin vuorovaikutus tapahtuu suoraan haastateltavan kanssa. Haastatteluissa etuina on joustavuus aineistoa kerätessä. (Hirsijärvi, Remes & Sajavaara. 2007, 199.) Kvalitatiivisessa tutkimuksessa haastattelujen tekeminen on toiminut päämenetelmänä. Haastattelun ei tarvitse alkaa tutkimuksen alusta, jolloin aineistoa voidaan kerätä joustavasti haastateltavan jaksamisen mukaan. (Hirsijärvi, Remes & Sajavaara. 2007, 200.) Kyselyn ja haastattelun avulla saadaan tietää, kuinka henkilöt ajattelevat, tuntevat ja uskovat. Kuitenkaan ne eivät kerro millä tavalla tutkittavat havaitsevat mitkä ympärillä tapahtuu. Havainnoinnin (observation) avulla saadaan selville tekevätkö ihmiset niin kuin he sanovat toimivansa. Tieteellinen havainnointi on tarkkailua. Havainnointi ei ole pelkästään näkemistä. Havainnointi on hankala menetelmä ja sen takia kysely ja haastattelu ovat osittain syrjäyttäneet sen. (Hirsijärvi, Remes & Sajavaara. 2007, 208.)

## 6 Toimintaympäristön käyttöönotto

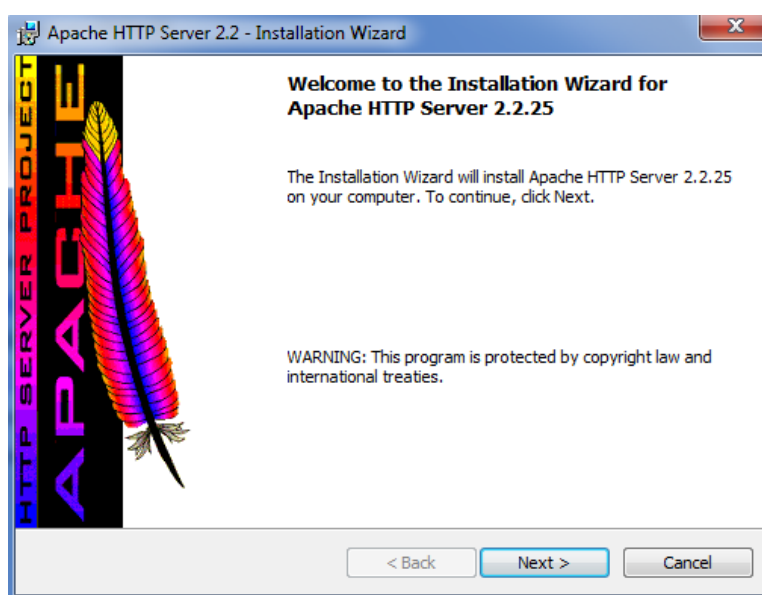
Toimintaympäristön käyttöönotto alkaa Apache-verkkopalvelimen asennuksella. Asennuksessa haetaan Internetistä asennettava ohjelma, joka asennetaan tietokoneelle. Apachen httpd.conf-tiedostoon tehdään tarvittavat muutokset, jotta voidaan aktivoida PHP-moduuli Apache-palvelimelta. PHP:n käyttöönotto alkaa hakemalla Internetistä PHP:n tarvitsemat dll-tiedostot. "php.ini"-tiedostoon tehdään tarvittavat muutokset, jotta PHP toimisi Apache-verkkopalvelimen kanssa. SQLITE3-tietokanta haetaan myös Internetistä. SQLITE3-tiedostot puretaan samaan kansioon, missä Apache-verkkopalvelimen Internet tiedostot sijaitsevat. CSS-tyylitiedostot tehdään omaan kansioon, josta Internet-sivut löytävät oikeat asetukset.

### 6.1 Apache-palvelimen asentaminen ja käyttöönotto

Apache-palvelimen asennus aloitetaan menemällä Internetiin kirjoittamalla Googlen haku-kenttään "apache download". Sivustolta valikoidaan osoite <http://httpd.apache.org/download.cgi>. Asennuksessa käytetään **Legacy Release** versiota, joka asennus hetkellä oli [2.2.27](#) (released 2014-03-26) Asennus jatkuu menemällä kohtaan **Binaries**, josta edelleen siirrytään kansioon nimeltä **win32**, josta löytyy 32- ja 64-bittiselle

Windows käyttöjärjestelmälle sama asennuspaketti [httpd-2.2.25-win32-x86-no\\_ssl.msi](httpd-2.2.25-win32-x86-no_ssl.msi). Asennuspaketti tallennetaan sille nimettyyn kansioon C-asemalle, josta se löytyy tarpeen tullen uudestaan, jos asennus menee pieleen. Samalla tehdään Apache palvelimelle C-aseman juureen uusi kansio myöhempää käyttöä varten, koska asennus tullaan kohdistamaan nimenomaan tähän kansioon.

Varsinainen Apache-palvelimen asennus aloitetaan aukaisemalla ladattu [httpd-2.2.25-win32-x86-no\\_ssl.msi](httpd-2.2.25-win32-x86-no_ssl.msi) asennuspaketti, jonka jälkeen aukeaa kuvan 1 mukaisesti installation wizard asennuksen käyttöliittymä. Käyttöliittymässä ei ole paljon vaihtoehtoja, joten siirrytään asennuksessa eteenpäin painamalla next-painiketta.



Kuva 1: Apache palvelimen-asennuksen aloituskuva

Apache palvelimen asennuksen lisenssiehdot tulee hyväksyä, jotta asennuksessa pääsee eteenpäin. Hyväksytään aktiiviseksi kohta I accept the terms in the license agreement, jonka jälkeen painetaan next-painiketta.

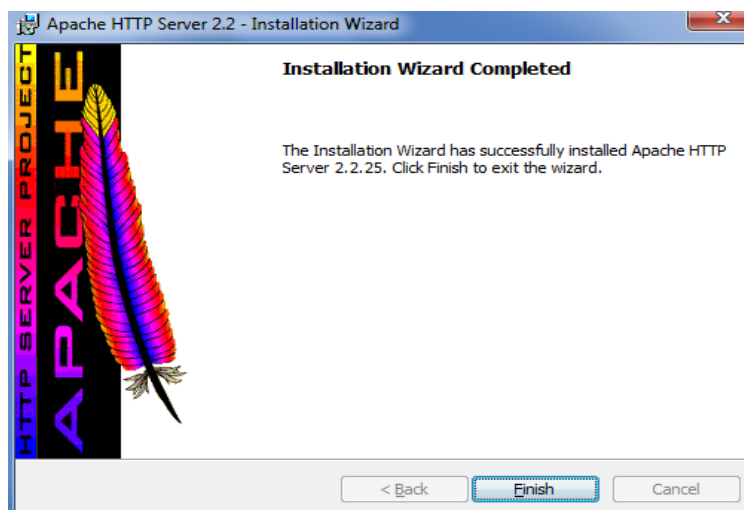
Kun lisenssiehdot on hyväksytty, niin sen jälkeen aukeaa sivu Read this first, jossa kerrotaan dokumentaatiosta, asennuksesta ja lisenssistä tietoja lyhyesti. Kustakin aiheesta löytyy Internet-linkki, mikäli haluaa saada lisätietoa kyseisestä asiasta. Asennuksessa mennään eteenpäin painamalla next-painiketta.

Tässä vaiheessa Apache-palvelimen asennus etenee samoin, kuin mikä tahansa Windows-koneelle asennettava ohjelmisto. Read this first sivuston jälkeen aukeaa Server information sivu. Tällä sivulla täytetään Network domain- Server name- ja Administrator email address tekstikentät. Mikäli tämä asennus tulisi julkiseen Internet-verkkoon, niin edellä mainittuihin

nimiin pitäisi laittaa viralliset tiedot. Mutta koska tämä asennus tulee olemaan omassa sisä-verkossa WLAN-reitittimen avulla ilman Internet-yhteyttä, niin nimet voivat olla epävirallisia nimiä. Lisäksi on valittavana kaksi kohtaa jossa toisessa sanotaan for all users, on Port 80 as a Service - recommended. Tämä tarkoittaa sitä, että kaikki tietokoneen käyttäjät voivat käyttää Apache palvelinta portissa 80 palveluna. Eli kun tietokone käynnistetään, niin Apache palvelin käynnistyy automaattisesti. Sivustossa voidaan valita myös kohta Only for the current user on port 8080 when started manually. Kyseisen asennuksen suorittanut käyttäjä pystyy käyttämään Apache-palvelinta vain portissa 8080. Lisäksi palvelin pitää käynnistää manuaalisesti. Asennuksessa valitaan kuitenkin kohta recommended ja mennään asennuksessa next-painiketta. Server informaation-sivun jälkeen aukeaa Setup type-sivu, jolloin tässä asennuksessa valitaan kohta Typical, koska Typical sisältää kaiken tarvittavan palvelinasennukseen. Asennuksessa siirrytään eteenpäin painamalla next-painiketta.

Setup type-sivun jälkeen aukeaa asennuksen kannalta tärkeä Destination folder-sivu. Tällä sivulla etsitään aikaisemmin luotu apache kansio C-asemalta ja painetaan ok-painiketta. Tämä valinta johtuu pelkästään Windows 7-käyttöjärjestelmästä, mikä on myös asennusalueena. Oletuksena ohjelma ehdottaa asennusta Program folder x86-kansioon C-asemalle. Windows 7-käyttöjärjestelmä suojaa tuon kyseisen kansion, eikä anna luoda uusia tiedostoja, vaan pelkkiä kansioita. Asetuksia pystyy kyllä muuttamaan, mutta niitä ei tarvitse tässä asennuksessa tehdä, koska asennuskansio on muutettu. Asennuksessa mennään eteenpäin painamalla ok-painiketta.

Setup type-sivun jälkeen aukeaa Ready to install the program-sivu. Asennus suoritetaan loppuun painamalla install-painiketta. Mikäli vielä olisi tarvetta muuttaa edellä tehtyjä asetuksia, niin back-painiketta painamalla voi mennä takaisin edellisiin asetuksiin. Setup type-sivun jälkeen aukeaa "Installation wizard Completed"-sivu mikä tarkoittaa, että asennus on suoritettu onnistuneesti loppuun. Installation wizard Completed-sivun voi sulkea painamalla finish-painiketta kuvan 2 mukaisesti.



Kuva 2: Installation wizard Completed sivu

## 6.2 Apache-palvelimen testaaminen

Käynnistämällä Internet-selaimen ja kirjoittamalla selaimeen localhost, pitäisi ruudulle ilmes-  
tyä **It works!**-teksti. Tämä kertoo sen, että palvelin on toiminnassa, koska localhost käsky  
toimii. Mikäli tekstiä ei ilmestynyt selaimeen, niin silloin olisi hyvä tarkistaa paikallinen tieto-  
turvaohjelmisto, mikä saattaa estää palvelimen porttien käytön. Jos kyseinen portti on jo  
käytössä pitäisi selvittää, mikä muu ohjelma mahdollisesti käyttää samaa porttia.

## 6.3 PHP:n asentaminen ja käyttöönotto

Pelkästään Apache-palvelimen asennus ei mahdollista PHP:n automaattista toimintaa, vaan  
myös PHP täytyy ladata Internetistä. PHP:lle luodaan oma kansio "C:\php". Php-kansioon tal-  
lennetaan kaikki PHP:n asennustiedostot. PHP:n lataaminen onnistuu osoitteesta  
<http://windows.php.net/download/>.

PHP:n Internet-sivustolla <http://php.net/manual/en/install.windows.manual.php> oli asen-  
nushetkellä seuraavanlainen huomautus, mikä on myös hyvä huomioida ennen asennusta.  
PHP-tarvitsee toimiakseen VC9-version, joko Microsoft 2008 C++ Runtime (x86) tai Microsoft  
2008 C++ Runtime (x64) ohjelman riippuen asennettavasta käyttöjärjestelmästä.

Asennettavassa tietokoneessa on jo entuudestaan käytössä Microsoft 2008 C++ Runtime paket-  
tiohjelma. Oikean version löytäminen tältä Internet sivustolta ei välttämättä ole helppoa,  
mutta kun selataan sivustoa, niin sieltä löytyi versio VC9 x86 Thread Safe (2014-Aug-15  
20:02:15), josta valitaan zip-asennus, joka tallennetaan ja puretaan "C:\php"-kansioon. Pur-  
kamisen voi tehdä esim. 7-zip ohjelmalla, mikä löytyy osoitteesta  
<http://www.7-zip.org/download.html>.

Tämän jälkeen aukaistaan "C:\php"-kansio, josta etsitään php.ini-production tiedosto. php.ini-production tiedosto on hyvä varmuuden vuoksi kopioida, koska toinen tiedostoista nimetään uudestaan php.ini nimiseksi. Kun php.ini on olemassa, niin httpd.conf tiedostoa pitää käydä muokkaamassa. Muokkaus suoritetaan tekstieditorilla, esim. Windows Note ohjelmalla. Muokkauksessa on kuitenkin käytetty Note++-ohjelmaa, jonka löytää osoitteesta <http://notepad-plus-plus.org/>. Muokkaus aloitetaan aukaisemalla kansiota C:\Apache\conf, jolloin aukaistaan httpd.conf- tiedosto, joka sisältää Apache-palvelimen kaikki asetukset mistä PHP pitää käydä aktivoimassa. Httpd.conf-tiedostoon lisätään seuraavat rivit viimeisen **#LoadModule** tekstirivin jälkeen, mikä löytyy httpd.conf-tiedostosta kuvan 3 mukaisesti, jonka jälkeen tallennetaan ja suljetaan httpd.conf-tiedosto.

```
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so

# -----PHP MODULLI -----
LoadModule php5_module "c:/php/php5apache2_2.dll"
AddHandler application/x-httpd-php .php

# configure the path to php.ini
PHPIniDir "c:/php"
```

Kuva 3: PHP:n aktivointi httpd.conf-tiedostossa

Lopuksi lisätään Windows ympäristömuuttujiin kohdasta ohjauspaneeli -> järjestelmä ja suo-  
jaus -> järjestelmä ->, järjestelmän lisäasetukset ja lisäasetukset välilehdeltä kohta ympäris-  
tömuuttujat. Valitaan listalta kohta uusi jolloin muuttujan nimeksi kirjoitetaan path ja muut-  
tujan arvoksi kirjoitetaan C:\php, jonka jälkeen kaikki PHP:n tarvittavat asetukset on määri-  
tetty.

#### 6.4 SQLITE3:n asentaminen ja käyttöönotto

SQLITE3 pitää myös käydä lataamassa Internetistä osoitteesta <http://www.sqlite.org/download.htm>. Sivustolla etsitään kohta Precompiled Binaries for Windows, josta valikoidaan sqlite-dll-win32-x86-3080600.zip-tiedosto. Tallennetaan zip-tiedosto omalle koneelle ja puretaan zip-tiedoston sisältö kohtaan C:\Apache\htdocs\kotisivu. kotisivu-kansio tulee myöhemmin sisältämään kaikki HTML- ja PHP-tiedostot. ZIP-tiedoston sisältö on **sqlite3.dll** ja **sqlite3.def** nimiset tiedostot. Nämä kaksi tiedostoa ohjaavat SQLITE3-tietokanta toimintoja, jotta PHP kykenee lukemaan ja kirjoittamaan SQLITE3-tietokantaan. Apache ja PHP eivät suoraan toimi SQLITE3-tietokannan kanssa ja ne pitää käydä aktivoimassa php.ini-tiedostosta. Php.ini-tiedosto löytyy kohdasta C:\php\php.ini.

Etsitään php.ini-tiedostosta kohta "Directory in which the loadable extensions (modules) reside". Tämän alapuolella lukee On Windows, jonka seuraavalle riville kirjoitetaan samalla tavalla "extension\_dir = c:\php\ext" kuvan 4 mukaisesti.

```
; Directory in which the loadable extensions (modules) reside.
; http://php.net/extension-dir
; extension_dir = "./"
; On windows:
extension_dir = "c:\php\ext"
```

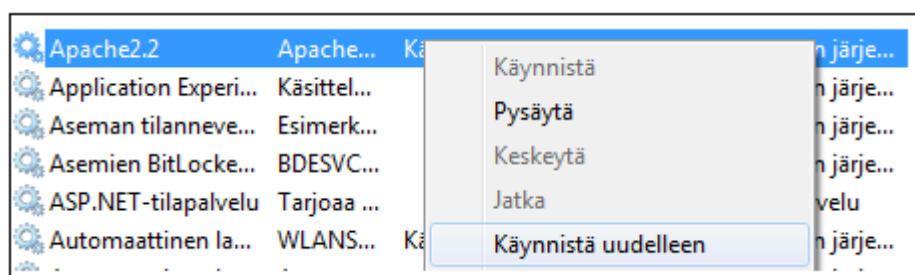
Kuva 4: extension\_dir asetuksen muuttaminen

Php.ini-tiedostosta pitää etsiä kohta Windows Extensions, jonka alapuolella on extension=php\_sqlite3.dll. Tämän rivin alusta poistetaan puolipiste kuvan 5 mukaisesti, joka aktivoi php\_sqlite3.dll-rivin. Kun muutokset on tehty, niin php.ini tiedosto voidaan tallentaa.

```
; Windows Extensions
;
;extension=php_bz2.dll
;extension=php_curl.dll
extension=php_sqlite3.dll
```

Kuva 5: Windows-Extensions asetusten muuttaminen

Kuvan 6 mukaisesti Apache-palvelin pitää käynnistää uudestaan, jotta uudet asetukset tulevat voimaan. Eri Windows käyttöjärjestelmäversioissa löytyy eri tavalla Windows-palvelut, mutta Windows 7-käyttöjärjestelmässä kirjoittamalla Windowsin etsi-toiminnolla sanan palvelut ja uudelleen käynnistämällä "Apache2.2" ohjelman.



Kuva 6: Apache2.2 uudelleen käynnistys Windows-palveluista

## 6.5 PHP:n ja SQLITE3:n tarkistus

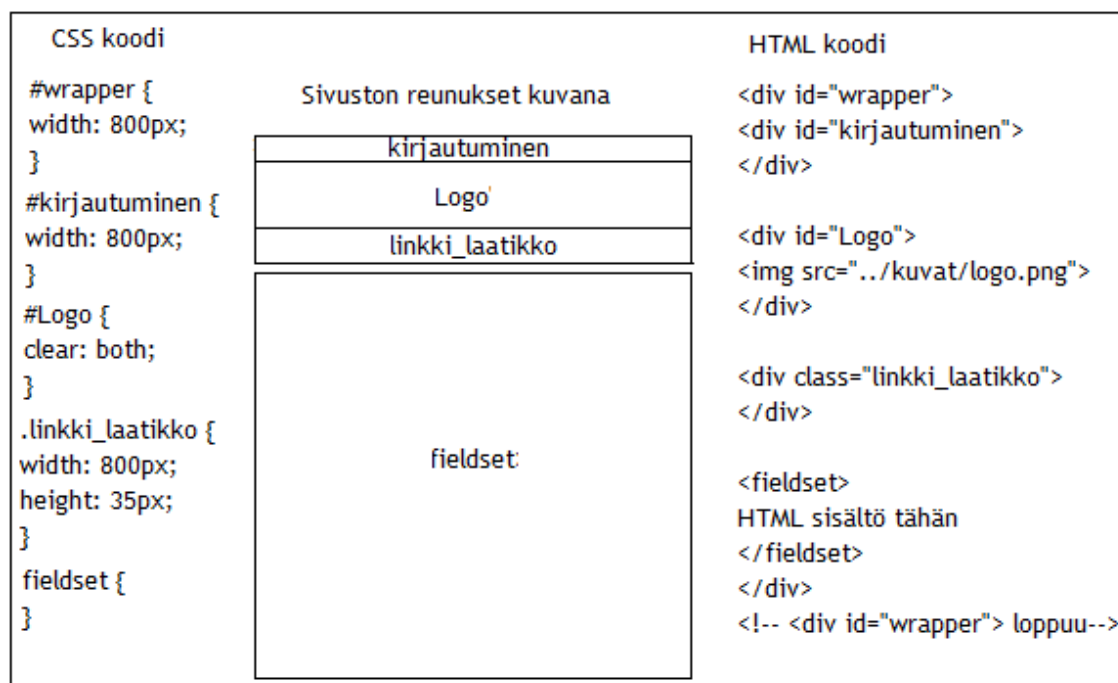
PHP voidaan testata siten, että mennään C:\Apache\htdocs kansioon, johon luodaan uusi kansio nimeltä testi. Testi nimiseen kansioon tehdään uusi tiedosto nimeltä testi.php. Tiedoston sisältö on seuraavanlainen `<?php phpinfo(); ?>` testi.php-tiedosto tallennetaan ja aukaistaan Internet selaimella localhost/testi/testi.php. Selaimeen aukeaa kuvan 7 mukainen PHP-logo ja erilaista tietoa Apachen, PHP:n ja SQLITE3:n tiedoista, sekä monista muista asetuksista.



Kuva 7: phpinfo ohjelman logo

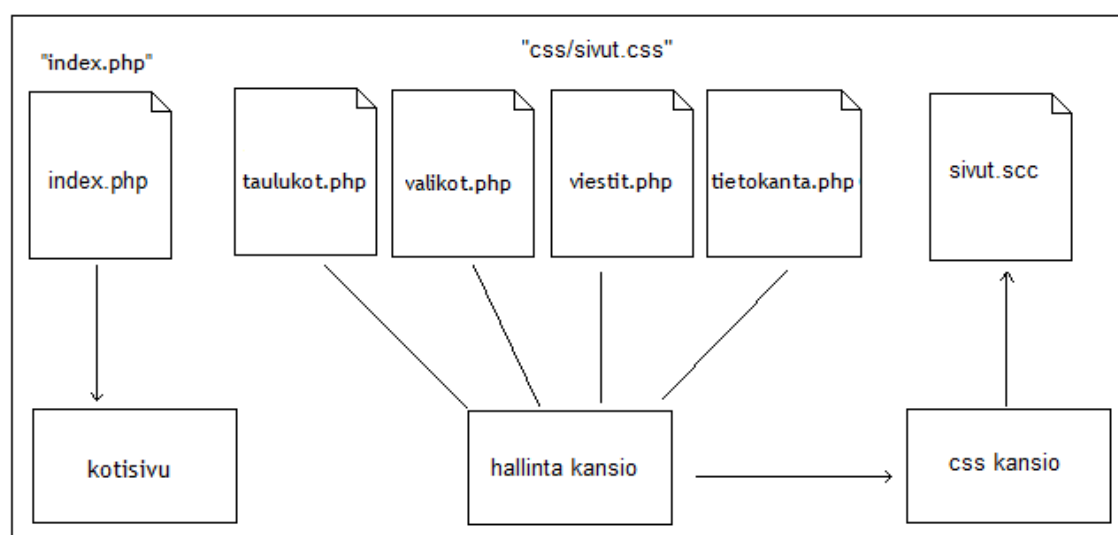
## 6.6 Verkkosovelluksen sivujen rakenne CSS tyylitiedostoilla

Kaikki ohjelman sivustot koostuvat eri CSS-elementistä kuvan 8 mukaisesti. Elementti voi olla leveydeltään esim. 800-pixeliä ja korkeudeltaan 600-pixeliä koko Internet-sivuston koosta. Tällöin tämä elementti varaa tuon kyseisen alueen, jolloin sisälle voi upottaa muita elementtejä. Elementit ovat myös laatikkomaisia alueita. Käytettävyys Internet sivustolla on tärkeä asia. Tähän voidaan vaikuttaa wrapper-elementillä, minkä tehtävä on pitää Internet sivuston koko sisältöä keskiosassa. Ilman wrapper-elementtiä sivuston koko sisältö olisi laidassa, mikä ei näyttäisi ulkoisesti kovin hyvältä. Elementit on hyvä nimetä niin, että niiden tarkoituksen tietää heti. Kirjautuminen-elementti pitää sisällään ainoastaan etusivulla olevan kirjautumiseen varatun alueen. Logo-elementti pitää sisällään nimensä mukaisesti yrityksen logon, mikä sijoitetaan tämän elementin sisälle. Linkki\_laatikko sisältää linkki-navigaation ja ".linkki\_laatikko" ainoa tarkoitus on pitää HTML-linkit tämän class-valitsimen sisällä. fieldset-toiminnon tarkoitus on sisältää varsinainen HTML-sisältö fieldset-reunojen sisällä. Lisäksi fieldsetin koko vaihtuu automaattisesti sisällön mukaan. Ei riitä, että luodaan pelkkä CSS-tyylitiedosto sillä samat elementtien nimet pitää löytyä myös varsinaisen HTML-sivun koodista.



Kuva 8: CSS- ja HTML-koodi käyttöliittymän reunukset

CSS-tyylitiedostot on linkitetty kaavion 1 mukaisesti sivut.css-nimiselle sivulle. Index.php-sivu sisältää omat asetukset, joita ei ole linkitetty. Linkitys tarkoittaa sitä, että varsinaisen Internet sivuun on kerrottu paikka mistä sen sivun tyylitiedostot haetaan eikä tyylitiedostoja tarvitse tällöin sisällyttää varsinaiselle Internet-sivulle. Linkityksen etuna on myös se, että mikäli myöhemmin tarvitsisi tehdä muutoksia tyylitiedosto asetuksiin, niin se onnistuisi helposti yhdestä paikasta.



Kaavio 1: CSS-tyylitiedostojen sijainnit



Tyylejä on mahdollista laittaa suoraan sivuston ”head” osioon laittamalla ne style-tagien sisään. Tämä toimintatapa ei kuitenkaan toimi parhaalla mahdollisella tavalla lisätä tyylejä dokumenttiin. Mikäli tuodaan toisen sivun tietoja käyttäen samoja tyylejä, pitäisi tällöin luoda kaksoiskappale CSS:stä uudelle sivulle. Jos kuitenkin haluaisi muuttaa tyyliä, se pitäisi toteuttaa niin monessa monta kertaa kuin on sitä sivustoilla käyttänyt yhden sijasta. CSS mahdollistaa tyylien säilyttämisen yhdessä tai useammassa erillisessä tyylitiedostossa. On olemassa kaksi tapaa liittää tyylitiedosto sivustolle. Toinen tavoista on linkittää tai tuoda dokumentin sisään olemassa olevat tiedot tyyleistä.” (Collison, Budd & Moll 2006, 20.) Sivustoissa on käytetty molempia tapoja lukea CSS-tyylitiedostoja.

## 6.7 Tietokannan ja käyttöliittymän käyttöönotto

Vaatimuksien määrittelyssä mietitään mallinnettavaa yritystä, haastatellaan kaikkia mahdollisia osapuolia, joiden sisältämä tieto auttaa selvittämään mahdolliset ongelmat nykyisen järjestelmän ja tulevien tarpeiden analysoimiseksi. Lisäksi selvitetään kaikki toiminnalliset tietotarpeet tulevaa tietokantaa varten. Olemassa olevasta tiedosta muotoillaan itse tietokantarakenne. Tässä voidaan käyttää muotoilumenetelmää, kuten yhteyksikaavioiden ER-kaavioiden tekemistä, sisällön merkitysten muotoilua tai tehtävien muotoilua. Kaikki tieto mitä muotoilumenetelmien avulla voidaan esittää graafisesti voidaan näyttää tietokannan eri osa-alueita, kuten taulut ja taulujen yhteydet, sekä yhteyksien ominaisuudet. (Hernandez 2000, 28.)

Haastattelut toimivat tärkeänä osana koko projektin aikana tietokannan suunnitteluprosessia ja niitä tehdään projektin tärkeissä kohdissa. Jos mietitään, että olet suunnittelemassa tietokantaa, joka toimii haluamallasi tavalla ja tukee koko työyhteisön tuottamaa työtä, silloin haastatteluja tehdään koko ajan. Tämä tarkoittaa, että pitää olla koko suunnitteluprosessin ajan tekemisissä avainhenkilöiden kanssa ja (organisaation koosta riippuen) itse omistajan kanssa. Jos työskentelet pienessä organisaatiossa, jossa on muutama henkilö tai olet tekemässä tietokantaa vain itseäsi varten, tulet ”haastattelemaan itseäsi”. Toisin sanoen toimit sekä haastattelijana että haastateltavana. Sinä olet henkilö, joka vastaa kysymyksiin (Hernandez 2000, 72.)

Tietokannan valintaan on valittu kevyt tiedostopohjainen SQLITE3-tietokanta. Valintaan vaikutti se, että sitä ei tarvitse erikseen asentaa ja siinä toimivat samat tutut SQL-käskyt kuin MySQL-tietokannassa. Verkkosovelluksen tiedostot luovat automaattisesti kaikki käytettävissä olevat tietokannat, mikä on toteutettu käyttämällä PHP-ohjelmointia. Tietokannan suunnittelu on mietitty tarkkaan ensiksi paperilla ja siitä toteutettu PHP-koodiin tekstieditorilla.

### 6.7.1 Kysymysten ja vastauksien lisääminen tietokantaan

Käytettävyyttä voidaan testata monella eri tapaa. Esimerkiksi tieteellisesti oikeaksi todettuja ja periaatteellisen tiedon kerääminen testaamalla vaatii ymmärrystä ja erikoisaamista. Projekteja johtavat henkilöt eivät pidä myöhäisessä vaiheessa ilmenevistä muutostarpeista. Esimerkiksi valikkojen järjestyksen muuttaminen ei aiheuta isoa ongelmaa loppupuoella projektia. Isot käyttöliittymän toimintaa ja arkkitehtuuria sisältävät muutokset sen sijaan laittavat aikataulut ja budjetit sekaisin. (Wiio 2004, 218.)

Kuvassa 9 on oletus näkymä sivustolta, kun tietokantaan ei ole tallennettu asiakkaalle tarkoitettua kysymystä, eikä vastaukselle tarkoitettuja vaihtoehtoja. Verkkosovellus lukee tietokannasta sille määritetystä paikasta, että onko tietokantaan tallennettu oletusviestien tilalle uutta tietoa. Mikäli tietoa ei ole tallennettu verkkosovellus näyttää sen tiedon punaisella taustalla olevalla kirjoituksella tai ilmoittaa, että ”Asiakkaan kysymys puuttuu”. Sivustolla on tarkoitus saada yrityksen henkilökunnan huomio, koska verkkosovelluksen kysymykset asetetaan tietokantaan aina kyseistä asiakaspalautetapahtumaa varten.

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

*Profiili Palvelut Oy*

PALAUTE

**Oletustekstiä ei ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.**

**Oletustekstiä ei ole muutettu hallintasivuilla.**

Asiakkaan kysymys 1 puuttuu  
▼

Asiakkaan kysymys 2 puuttuu  
▼

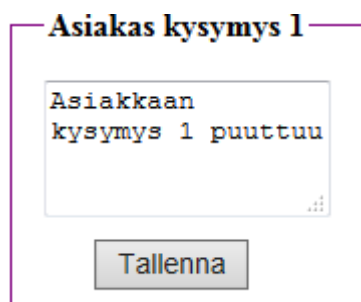
Asiakkaan kysymys 3 puuttuu  
▼

Asiakkaan kysymys 4 puuttuu  
▼

Asiakkaan kysymys 5 puuttuu  
▼

Kuva 9: Oletusnäkymä sivustolta ilman asetuksia

Asiakkaille tarkoitetut kysymykset ja vastaukset tallennetaan hallintasivustolla. Verkkosovellus lukee automaattisesti PHP:llä tietokannasta kuvan 10 mukaisesti tiedon olemassa olevasta kysymyksestä. Alas-vetovalikkokysymyksiä on yhteensä viisi ja yksi iso tekstikenttä. Kaikille kysymyksille on oma alue, josta kysymys asetetaan. Verkkosovellusta hallinnoivan ei tarvitse osata itse lisätä tietoja tietokantaan, vaan riittää että kysymys kirjoitetaan tekstikenttään ja painetaan tallenna-painiketta.



**Asiakas kysymys 1**

Asiakkaan kysymys 1 puuttuu

Tallenna

Kuva 10: Asiakaskysymyksen tallentaminen

Kuvan 11 mukaisesti on nähtävillä ”asiakas kysymys 1”, mikä on toteutettu HTML- ja PHP-koodilla. Tässä käytetään hyväksi PHP-funktiota tiedon lukemiseen tietokannasta tekstikentän sisälle. Joka kerta, kun sivustoa päivitetään, niin tämä kuvan 11 mukainen funktio lukee tiedon.

```
<textarea name="as_kys1_textbox" cols="15" rows="3">
  <?php
    $Asiakaskysymys1 = Asiakaskysymys1();
    echo $Asiakaskysymys1;
  ?>
</textarea>
```

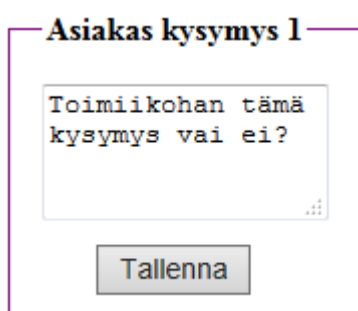
Kuva 11: HTML- ja PHP- koodi

Kuvan 12 mukaisesti on nähtävillä PHP-funktion sisältö, jossa tiedot haetaan tietokannasta. Funktiot voidaan pitää erillisellä sivulla, jolloin funktiota kutsutaan. Funktioilla on myös mahdollista selkeyttää olemassa olevaa sivuston koodirakennetta. Kuvassa 12 Asiakaskysymys1-funktiolla valitaan tietokannasta KYSYMYKSET-tietue tietokannasta KYSYMYKSET, missä ID-arvo on yksi.

```
function Asiakaskysymys1 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYS1, * FROM KYSYMYKSET WHERE ID=1');
echo $row;
}
```

Kuva 12: Asiakaskysymys 1 PHP-funktion sisältö

Kuvan 13 mukaisesti tallennetaan ensimmäisen kerran hallintasivuilla tietokantaan kysymys ”Toimiikohan tämä kysymys vai ei?”-tekstin. Asiakkaalle tarkoitettu ”kysymys 1” tallentuu aina edellisen kysymyksen päälle.



Kuva 13: Hallintasivustolla asetetaan kysymys asiakkaalle

Kuvan 14 mukaisesti ”kysymykset”-tietokantaan päivitetään ”kysymys1” kolumniin olemassa olevan tekstikentän tieto, missä tietokannan ID arvo on 1. Lisäksi PHP-koodin lopussa on selaimella päivitys-funktio, jonka tarkoituksena on päivittää tallennettu tieto näkyviin selaimen.

```
<?php
$as_kys1_textbox = $_POST['as_kys1_textbox'];
if(isset($_POST['nappula1_kys']))
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$db->exec("UPDATE KYSYMYKSET SET KYSYMYS1='$as_kys1_textbox' WHERE ID=1");
$db->close();

$selain_update = selain_update();
echo $selain_update;
}
?>
```

Kuva 14: PHP-koodilla päivitetään kysymys tietokantaan

Kuvan 15 mukaisesti tieto on tallennettu tietokantaan, jossa kukin kysymys tallennetaan aina omalle ID-riville. Myös kuvassa 15 on nähtävillä, kuinka ”kysymys 1” on muutettu, mutta ”KYSYMYS 2” kohdalla lukee edelleen ”Asiakkaan kysymys 2 puuttuu”.

ID	KYSYMYS1	KYSYMYS2
1	Toimiikohan tämä kysymys vai ei?	
2		Asiakkaan kysymys 2 puuttuu
3		
4		
5		

Kuva 15: Näkymä tietokannassa olevista tietueista

Hallintasivuilla kuvan 16 mukaisesti on samalla sivulla mahdollisuus testata kysymyksen ja vastauksen toimintaa. Siinä näkee valikon järjestyksen ja sen, että onko asiakkaalle tulevat kysymykset ja valikot oikeassa järjestyksessä. ”Toimiikohan tämä kysymys vai ei” on onnistuneesti luettavissa käyttöliittymällä tietokannasta, mutta asiakkaalle esitettävät valikon vastaukset vielä puuttuvat.

**Kysymys 1 valikon testaaminen**

Toimiikohan tämä kysymys vai ei?

Kuva 16: Kysymyksen 1 testaaminen hallintasivustolla

Kuvan 17 mukaisesti hallintasivuilla asetetaan ”asiakaskysymys 1”-valikon vastaukset. Kuvassa 17 asetetaan ”Testi vastaus”-teksti, jolla voidaan todeta tämän kohdan toimivuus. Tallennus tapahtuu painamalla tallenna-painiketta.

**Kysymys1 vastauksen lisääminen**

Testi vastaus

Tallenna

Kuva 17: Asiakkaalle lisättävän vastauksen lisääminen valikkoon

Kuvan 18 mukaisesti on nähtävillä PHP-koodi, jolla tallennetaan tietokantaan ”Testi vastaus”-kirjoitus. Lisäksi tallennettu tieto pitää näkyä heti selaimessa ja ilman selaimen päivitystä sivuston näkymä ei olisi reaaliaikainen.

```

<?php
if (isset($_POST['nappula1_lis']))
{
    $textareal_lis = $_POST['textareal_lis'];
    $db = new SQLite3('../tietokanta/valikko1_lisaa.db');
    $db->exec("INSERT INTO VALIKKO1_LISAA (VALIKKO) VALUES
    ('$textareal_lis')");
    $db->close();

    $selain_update = selain_update();
    echo $selain_update;
}
?>

```

Kuva 18: PHP-koodi valikkoon lisättävästä vastauksesta tietokantaan

Mikäli käyttäjällä on olemassa aikaisempaa tietämystä tietokoneiden käytöstä, hän pystyy mahdollisesti tutkimaan sovelluksesta tarvitsemiaan toimintoja mm. valikoista ja työkalupalkeista. Uteliaammat käyttäjät pyrkivät kokeilemalla tuottamaan itselleen käsitystä siitä, mikä on mahdollista ja mikä ei. He yrittävät muodostaa käsitystä siitä, mistä he löytävät tarvitsemansa tiedot ja toiminnot käyttöliittymästä. (Wiio 2004, 153.)

Sovelluksen pitää ilmoittaa käyttäjälle olemassa olevat toimintavaihtoehdoistaan esimerkiksi valikkojen avulla (Wiio 2004, 145.). Kuvan 19 mukaisesti on nähtävillä SQLITE3-tietokantaan asetettu ”Testi vastaus”-teksti, jossa käytetään automaattista AUTOINCREMENT-arvoa, joka on numeraalinen ja tämän tietokannan pääavain. Lisäksi AUTOINCREMENT-arvo lisää automaattisesti seuraavan numeron tietokantaan. PHP-koodiin on kirjoitettu ”ID 1” ja ”valikko 1” kohtaan arvo ”Valitse”. Tämä on tehty siksi, että ohjelman käyttöliittymän valintapainikkeen teksti on aina samanlainen eikä ylläpitäjän tarvitse huolehtia siitä. Lisäksi tämä vaikuttaa AUTOINCREMENT-arvoon, mikä alkaa seuraavasta numerosta ”2”. Ennen kuin asiakas aloittaa monivalintakysymyksiin vastaamisen on valikosta nähtävillä arvo ”Valitse”. Tämä toiminto myös ohjaa asiakasta valitsemaan vastauksen ilman, että siinä lukisi heti jokin vastaus tai että kohta olisi tyhjä. Sovelluksen pitää ilmoittaa käyttäjälle olemassa olevat toimintavaihtoehdoistaan esimerkiksi valikkojen avulla. (Wiio 2004, 145.)

ID	VALIKKO
1	Valitse
2	Testi vastaus

Kuva 19: SQLITE3-tietokannan vastausvaihtoehdot

Verkkosovellus säästää edellisen kertojen vastauksien vaihtoehdot kuvan 20 mukaisesti, joita voi käyttää uudestaan tai luoda uusia vastaus vaihtoehtoja. ”kohdassa Kysymys 1 vastauksien varasto” on mahdollista siirtää vastaus painamalla lisää-painiketta, minkä jälkeen vastaus siirtyy ”Kysymys 1 valikon järjestys” kohtaan. ”Kysymys 1 valikon testaaminen”-kohdassa voi tarkistaa alasvetovalikon järjestyksen ennen kuin sivusto voidaan antaa asiakkaiden käyttöön. Mikäli alasvetovalikon järjestys on väärä, niin väärän vastauksen voi poistaa poista painikkeella.

**Kysymys 1 vastauksien varsto**

ID	VALIKKO	TOIMINNOT	
1	Valitse	poista	Lisää
5	Ystävällistä	poista	Lisää
7	Ammattitaitoista	poista	Lisää
8	Epäasiallista	poista	Lisää
10	Välinpitämätöntä	poista	Lisää
11	Testi vastaus	poista	Lisää

**Kysymys 1 valikon järjestys**

ID	VALIKKO	TOIMINNOT
1	Valitse	poista

**Kysymys 1 valikon testaaminen**

Asiakkaan kysymys 1 puuttuu

Valitse ▼

Kuva 20: Alasvetovalikon vastaukset

Kuvan 21 mukaisesti on PHP-koodilla toteutettu molemmille painikkeille omat koodit ”poista”- ja ”Lisää”-painikkeet. PHP:ssä niille on omat nimet, mitkä PHP tunnistaa, kun painiketta painetaan. Asiakas ei voi nähdä missään vaiheessa verkkosovelluksen toiminteita. Lisäksi ohjelmoinnissa on tärkeitä yrittää pitää koodin sisältö vastaamaan ulkoasun nimiä, mikä helpottaa myöhempää käyttöä.

```

<?php
if (isset($_POST['valikko1_lisaa_poista']))
{
    $valikko1_lisaa_id = $_POST['valikko1_lisaa_id'];
    $db = new SQLite3('../tietokanta/valikko1_lisaa.db');
    $db->exec("DELETE FROM VALIKKO1_LISAA WHERE ID='$valikko1_lisaa_id'");
    $db->close();

    $selain_update = selain_update();
    echo $selain_update;
}

if (isset($_POST['valikko1_lisaa']))
{
    $valikko1_lisaa_id = $_POST['valikko1_lisaa_id'];
    $valikko1_lisaa_kysymys = $_POST['valikko1_lisaa_kysymys'];

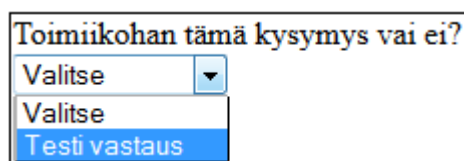
    $db = new SQLite3('../tietokanta/valikko1.db');
    $db->exec("INSERT INTO VALIKKO1 (ID,VALIKKO) VALUES
('$valikko1_lisaa_id', '$valikko1_lisaa_kysymys')");
    $db->close();

    $selain_update = selain_update();
    echo $selain_update;
}
?>

```

Kuva 21: PHP-koodi lisää- ja poista-painikkeista

Asiakas voi valita arvon "Valitse" ja se tulee näkymään silloin loppuraportissa arvona "Valitse". Tämä kertoo sen, että joko kysymys oli asiakkaalle vaikea tai siihen ei haluttu vastata. Sanan käyttö "Valitse" on kuitenkin paremman näköinen käyttöliittymässä, mikä myös ohjaa asiakasta valitsemaan jonkin arvon valikosta. Kuvan 22 mukaisesti on nähtävillä toimiva alasvetovalikolla toteutettu "testi vastaus"-teksti.



Kuva 22: Alasvetovalikon testaaminen

### 6.7.2 Verkkosovelluksen Ilmoitukset ja tietokanta

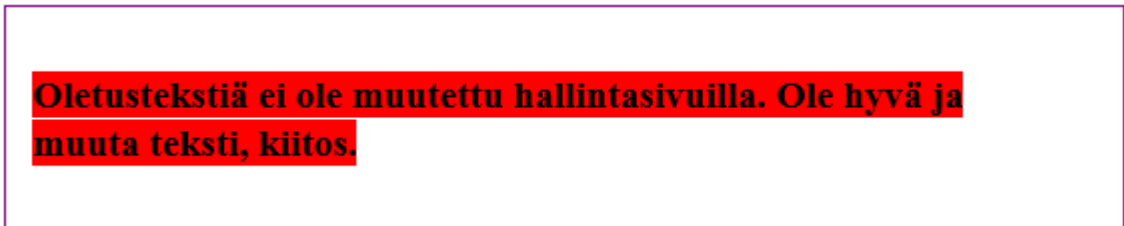
Mikäli pyrkii minimoimaan ajan käytön suunnitteluprosessissa, sitä suurempi riski on joutua tietokantojen kanssa ongelmiin. Edes tehokas tietokannan suunnitteluprosessi ei aina poista kaikkia ongelmia, joita voi tulla esille tietokantaa suunnitellessasi, mutta se mahdollistaa niiden merkittävän vähentämisen merkittävästi. (Hernandez 2000, 61.)



Silloin, kun asiakassivustolla ei ole tehty mitään muutoksia ohjelmaan, niin sivulla on nähtävillä eri kohdissa virheilmoituksia. Kun asiakas tulee ensimmäisen kerran sivustolle, niin asiakkaalle näytettävä tervetuloilmoituksen on oltava kunnossa. Ohjelman tervetuloilmoitus pitää voida muuttaa aina kyseisen tapahtuman mukaan.

Asiakkaille tarkoitetut kysymykset on sijoitettu käyttöliittymään omaan rajattuun laatikkoon, jotta käyttöliittymä olisi selkeämpi. Tämä on käyttäjäystävällinen toiminto ja graafisesti selkeämpi. Lisäksi otsikon tehtävänä on ohjata asiakasta vastaamaan oikealla tavalla. Asiakkaan sivuilla on myös yksi iso tekstikenttä mihin on tarkoitus kertoa vapaasti oma ehdotus palvelun kehittämisestä tai jostakin muusta asiasta. Lisäksi sovelluksen käyttöliittymässä on haluttu välttää tilannetta missä on ilmoitettu numeroin ja tekstillä päällekkäin ”täysin samaa mieltä”, jossa alhaalla on numero 1 ”jokseenkin samaa mieltä”, jossa alhaalla on numero 2 ”Neutraali kanta”, jossa alhaalla on numero 3. Käyttäjä menee sekaisin joutuessaan vertaamaan kahta eri arvoa numeraalisen ja tekstin väliltä. Vastauksesta voi tulla epävarma eikä käyttäjä välttämättä tiedä, että onko numero 1 parempi vastaus kuin numero kolme.

Kuvan 23 mukaisesti punaisella taustalla olevan tekstin tarkoitus on saada henkilökunta muuttamaan tietokannassa olevaa oletusviestiä, jotta sitä ei ole esillä asiakaspalautetapahtumassa. Oletusviestien on tarkoitus ohjata henkilökuntaa toteuttamaan asiakaskysymykset aina samalla tavalla. Tällä tavalla pystytään ohjaamaan henkilökuntaa toimimaan oikein.



**Oletustekstiä ei ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.**

Kuva 23: Tervetuloa oletusviesti punaisella taustalla

Kuvan 24 mukaisesti on nähtävillä, kuinka PHP tarkistaa tietokannasta, että onko henkilökunta käynyt tallettamassa uuden arvon tietokantaan. Jos uutta arvoa ei ole annettu, niin PHP käy lukemassa ”else”-käskyn ja jos arvo ”oletus”-teksti toteutuu, niin virheviesti luetaan selaimen. Kuvassa 24 oikean viestin näyttäminen on toteutettu Asiakkaan\_alkutervehdys-funktiolla.

```

else //arvo1
{
    $db = new SQLite3('tietokanta/viestit.db');
    $row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=1');

    if ($row == "oletus") // Arvo 2
    {
        .....
        echo '<p> <span class="virhe">Oletustekstiä ei
ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.</span></p>';
    }
    else // Arvo 2
    {
        .....
        $Tervehdys = Asiakkaan_alkutervehdys();
        echo $Tervehdys;
    } // Arvo 2
} // arvo1
$db->close();
?>

```

Kuva 24: Tervetuloaviesti punaisella taustalla PHP-koodina

Tervehdyksen muuttaminen alkaa kuvan 25 mukaisesti viestit hallintasivustolta. Alkutervehdys luetaan automaattisesti tietokannasta ja sitä voi muuttaa käyttöliittymästä "Alkutervehdys"-valikosta tallenna-painiketta painamalla. Jos alkutervehdystä ei ole muutettu, niin käyttöliittymässä on punaisella taustalla oleva teksti.

**Alkutervehdys**

Alkutervehdys  
asiakkaalle tähän

Kuva 25: Alkutervehdyksen muuttaminen

Kuvan 26 mukaisesti on nähtävillä alkutervehdys PHP-koodina, mikä on tulostettu tekstikentän sisälle. PHP-koodissa on hyödynnetty funktio-ominaisuutta, mitä voi käyttää monessa PHP-sivussa uudestaan ilman, että koko ajan täytyisi kirjoittaa sama PHP-koodi.

```
<textarea name="as_kys1_textbox" cols="15" rows="4"><?php
$Alkuteksti = Alkuteksti();
echo $Alkuteksti;
?></textarea>
```

Kuva 26: PHP-funktiolla luetaan tiedot tekstikentän sisälle

Kuvan 27 mukaisesti on nähtävillä PHP-funktio, jolla alkutervehdys luetaan VIESTI-tietokannasta. Viestin varsinainen sisältö tulostuu näytölle käskyllä "echo \$row" käskyllä. Kuvassa 27 luetaan VIESTI-tietueesta tietokannasta VIESTIT missä on ID-arvo yksi. Lukemisessa käytetään käskyä "querySingle", jolla voidaan luke yksittäinen tieto tietokannasta.

```
function Alkuteksti ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=1');
echo $row;
}
```

Kuva 27: PHP-funktion luku VIESTI-tietokannasta

Kuvan 28 mukaisesti on nähtävillä PHP-koodi kun tallenna-painiketta on painettu. Alkutervehdyksen teksti päivitetään tietokantaan, jolloin alkutervehdyksen oletusteksti muutetaan arvoksi "ei", kun arvo on "ei", niin asiakassivuston PHP-koodi ei laita sivulle punaista taustatekstiä. Lopuksi selain päivitetään, jotta muutetut arvot näkyvät alkutervehdys tekstikentässä.

```

<?php
    $as_kys1_textbox = $_POST['as_kys1_textbox'];
    if(isset($_POST['nappula1_kys']))
    {
        $db = new SQLite3('../tietokanta/viestit.db');
        $db->exec("UPDATE VIESTIT SET VIEСТИ='$as_kys1_textbox' WHERE ID=1");
        $db->close();

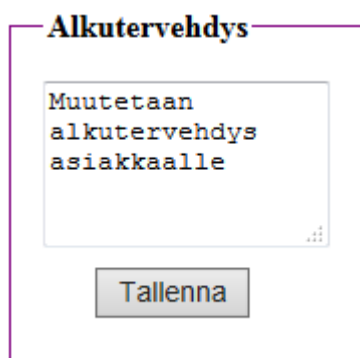
        $db = new SQLite3('../tietokanta/viestit.db');
        $db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=1");
        $db->close();

        $selain_update = selain_update();
        echo $selain_update;
    }
?>

```

Kuva 28: Alkutervehdyksen lukeminen PHP-koodilla tietokannasta

Kuvan 29 mukaisesti alkutervehdys muutetaan kohdasta "Alkutervehdys"-tekstikentässä. Muutettu tieto siirtyy tietokantaan, jolloin muut ohjelman toiminnot havaitsevat muutokset. Ohjelman käynnistyessä uudestaan luetaan tietokannassa olevat arvot myös uudestaan.



**Alkutervehdys**

Muutetaan  
alkutervehdys  
asiakkaalle

Tallenna

Kuva 29: Alkutervehdyksen muuttaminen hallintasivustolla

Kuvan 30 mukaisesti alkutervehdyksen muutos on nähtävillä "ID rivillä 1" kohdassa "viesti", missä ovat muut viestit, mitä ohjelmassa käytetään. Ohjelma muuttaa "oletus"-kohdan arvoksi "ei", jos henkilökunta on tehnyt ohjelmaan muutoksen.

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Palute viesti asiakkaalle tähän	oletus
3	Ylävalikon selitys asiakkaalle tähän	oletus
4	Alavalikon selitys asiakkaalle tähän	oletus
5	tyhjä	ei

Kuva 30: VIESTIT-tietokannan muutokset

Kuvan 31 mukaisesti alkutervehdys näkyy asiakassivustolla. Kuvasta 31 on nähtävissä, kuinka tärkeästä asiasta on kyse. Kun sivustolle saavutaan ensimmäisen kerran, niin siihen ei saa unohtaa oletusviestejä. Lisäksi samaan kohtaan alkutervehdyksen kanssa ilmoitetaan viesti asiakkaalle, kun asiakas on tallettanut tiedot yrityksen käytettäväksi. Tämä kiitos palautteesta-viesti pitää käydä muuttamassa tai siinä on punainen tausta samalla tavalla kuin valikoiden yläpuolella oleva kirjoitus on.

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

**ProfiPalvelut Oy**

**PALAUTE**

**Muutetaan alkutervehdys asiakkaalle**

**Oletustekstiä ei ole muutettu hallintasivuilla.**

Toimiikohan tämä kysymys vai ei?

Valitse

Kuva 31: Tallennettu uusi alkutervehdys asiakkaalle

Kuvassa 32 on nähtävillä se, että kiitosviestin tekstiä pitää muuttaa.

Tämä tapahtuu hallintasivuilla viestit-sivustolla kohdassa "Palaute viestistä".

Toiminto on ihan samanlainen, kuin oli alkutervehdyksessä, jossa kirjoitetaan viesti ja tallennetaan painiketta painamalla. Kiitosviesti tulee asiakkaalle esille, kun palaute on annettu.

Käyttäjänimi:  Salasana:   Kirjautunut: asiakas

**Profiili Palvelut Oy**

**PALAUTE**

**Kiitos palautteesta tekstiä ei ole muutettu hallintasivuilla.**

**Oletustekstiä ei ole muutettu hallintasivuilla.**

Asiakkaan kysymys 1 puuttuu

▼

Kuva 32: Kiitos palautteesta punaisella taustalla

Kuvassa 33 on nähtävissä ”palaute viestistä”- kenttä, missä annetaan uusi palaute tekstikenttään ja painetaan tallenna painiketta. Ilman muutosta punainen taustaväri näkyy selaimessa, joten tämä muutos on pakollinen tehtävä.

**Palaute viestistä**

Kiitos testi palautteesta

Kuva 33: Palauteviestin asettaminen tietokantaan

Kuvassa 34 on nähtävillä ”ID rivillä 2”-kohdassa ”viesti”, että ”Kiitos testi palautteesta” on tallennettu tietokantaan onnistuneesti. Myös kohdassa ”oletus”-arvo on muuttunut ”ei”-arvoksi.

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Kiitos testi palautteesta	ei
3	Ylävalikon selitys asiakkaalle tähän	oletus
4	Alavalikon selitys asiakkaalle tähän	oletus
5	tyhjä	ei

Kuva 34: Kiitos testi palautteesta tallennettu tietokantaan

Kuvassa 35 on nähtävillä ”Kiitos testi palautteesta” Internet-selaimessa. Ilmoitus tulee esille, kun asiakas on tallettanut asiakaspalautteen tiedot yrityksen käytettäväksi. Samassa kuvassa on nähtävillä punainen taustaväri valikoiden yläpuolella. Tämä pitää myös muuttaa hallinta-sivustolta.

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

**Profiili Palvelut Oy**

**PALAUTE**

**Kiitos testi palautteesta**

Oletus tekstiä ei ole muutettu hallinta sivuilla.

Toimiikohan tämä kysymys vai ei?

Valitse

Kuva 35: Kiitos viestistä ilmoitus ilman punaista taustaväriä

Kuvassa 36 on nähtävissä ”Ylävalikon teksti”-kenttä, missä annetaan uusi otsikko tekstikenttään ja painetaan tallenna-painiketta. Ilman muutosta punainen taustaväri näkyy selaimessa, joten tämä muutos on pakollinen tehtävä.

**Ylävalikon teksti**

Uusi testi  
kirjoitus ylä  
valikoille

Tallenna

Kuva 36: Ylävalikon tekstin muuttaminen

Kuvassa 37 on nähtävillä, että ”ID-rivillä 3” kohdassa ”viesti” on uusi ylävalikon teksti, sekä oletusteksti on muutettu arvoksi ”ei”. Arvo ”ei” pitää huolen siitä, että punaista taustaväriä ei näy asiakassivustolla.

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Kiitos testi palautteesta	ei
3	Uusi testi kirjoitus ylä valikoille	ei
4	Alavalikon selitys asiakkaalle tähän	oletus
5	tyhjä	ei

Kuva 37: Uusi testikirjoitus ylävalikolle tietokannassa

Kuvan 38 mukaisesti on nähtävillä, että asiakkaalle tarkoitetun vapaan tekstikentän otsikko on muuttamatta hallintasivuilla. Teksti pitää muuttaa samalla tavalla, kuin kaikki muutkin ilmoitukset tähän asti.

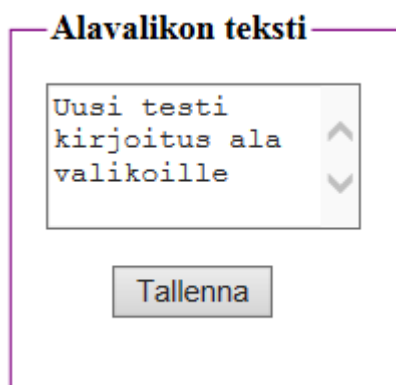
**Oletustekstiä ei ole muutettu hallintasivuilla.**

Tallenna

Kuva 38: Alavalikon oletusteksti punaisella taustavärillä



Kuvassa 39 on nähtävissä ”Alavalikon teksti”-kenttä, missä annetaan uusi otsikko tekstikenttään ja painetaan tallenna-painiketta. Ilman muutosta punainen taustaväri näkyy selaimessa, joten tämä muutos on pakollinen.



**Alavalikon teksti**

Uusi testi  
kirjoitus ala  
valikoille

Tallenna

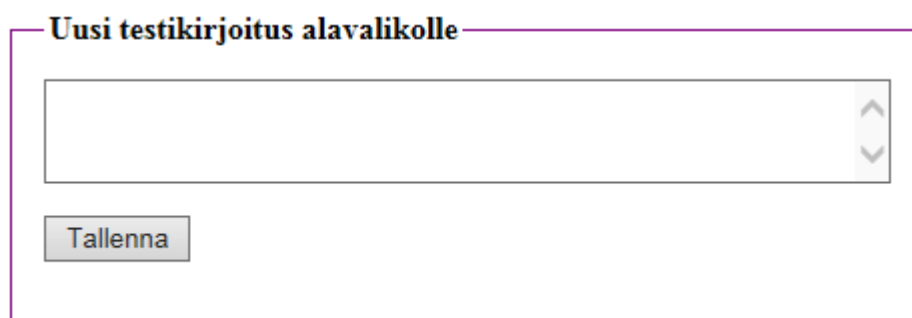
Kuva 9: Alavalikon tekstin muuttaminen

Kuvassa 40 on nähtävillä, että ”ID-rivillä 4” kohdassa ”viesti” on uusi alavalikon teksti sekä oletus teksti on muutettu arvoksi ”ei”. Arvo ”ei” pitää huolen siitä, että punaista taustaväriä ei näy asiakassivustolla.

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Kiitos testi palautteesta	ei
3	Uusi testi kirjoitus ylä valikoille	ei
4	Uusi testi kirjoitus ala valikoille	ei
5	tyhjä	ei

Kuva 40: VIESTI-tietokanta muutoksen jälkeen

Kuvan 41 mukaisesti on nähtävillä, että alavalikon otsikko ei ole enää punaisella taustavärillä. Myös tämän otsikon pitää olla hyvin kuvaava asiakkaalle, jotta asiakas saadaan kirjoittamaan tekstiä.



**Uusi testikirjoitus alavalikolle**

Tallenna

Kuva 41: Alavalikon oletusteksti on muutettu

## 7 Verkkosovelluksen käyttö tablet laitteilla

On mahdollista olettaa, että uuden sovelluksen käyttäjällä on tietokoneesta ja mahdollisesti muista mobiililaitteista tarvittavia yleistaitoja. Samoilla tietoteknisillä taidoilla käyttäjän pitäisi pystyä päättämään, kuinka hän pystyy tekemään haluamansa lopputulokseen. Käyttöliittymän pitää pystyä auttamaan käyttäjää selviämään kaikin mahdollisin tavoin. (Wiio 2004, 130.) Tämä asia on pyritty huomioimaan niin, että käyttäjä pystyy näkemään nopealla vilkaisulla ruudulta, ettei häneltä tulla vaatimaan ylivoimaisia tietoteknisiä taitoja. Lisäksi tablet-laitteiden valikot näkyvät samalla tavalla kuin minkä muun tahansa verkkokyselyn kysymykset. Tämä johtuu siitä, että sovellus noudattaa tablet-laitteella olevia käyttöliittymäasetuksia.

Kuvan 42 mukaisesti asiakassivusto on pääsääntöisesti tarkoitettu käytettäväksi tablet-laitteilla. Tablet-laitteita voi olla käytössä yhtäaikaaisesti useampi, joten tiedonkerääminen ei rajoitu pelkästään yhteen laitteeseen.

**PALAUTE**

**Palautetta antamaan palautetta**

**Palveluvalikko**

Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?

Ammattitaitoista

Asiakkaan haluamia tuotteita?

on saatavilla, hinnat merkitty oikein ja täsmällisesti

Yrityksen luotettavuuden vaikuttaa?

Valitse

Ongelmatilanteissa yrityksen työntekijällä?

Valitse

Tullessani yrityksen tiloihin haluaisin huomauttaa?

Valitse

**Mitä palvelua sinä muuttaisit paremmaksi?**

Haluaisin lisää ammattikirjallisuutta

Edel.    Seur.

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p å

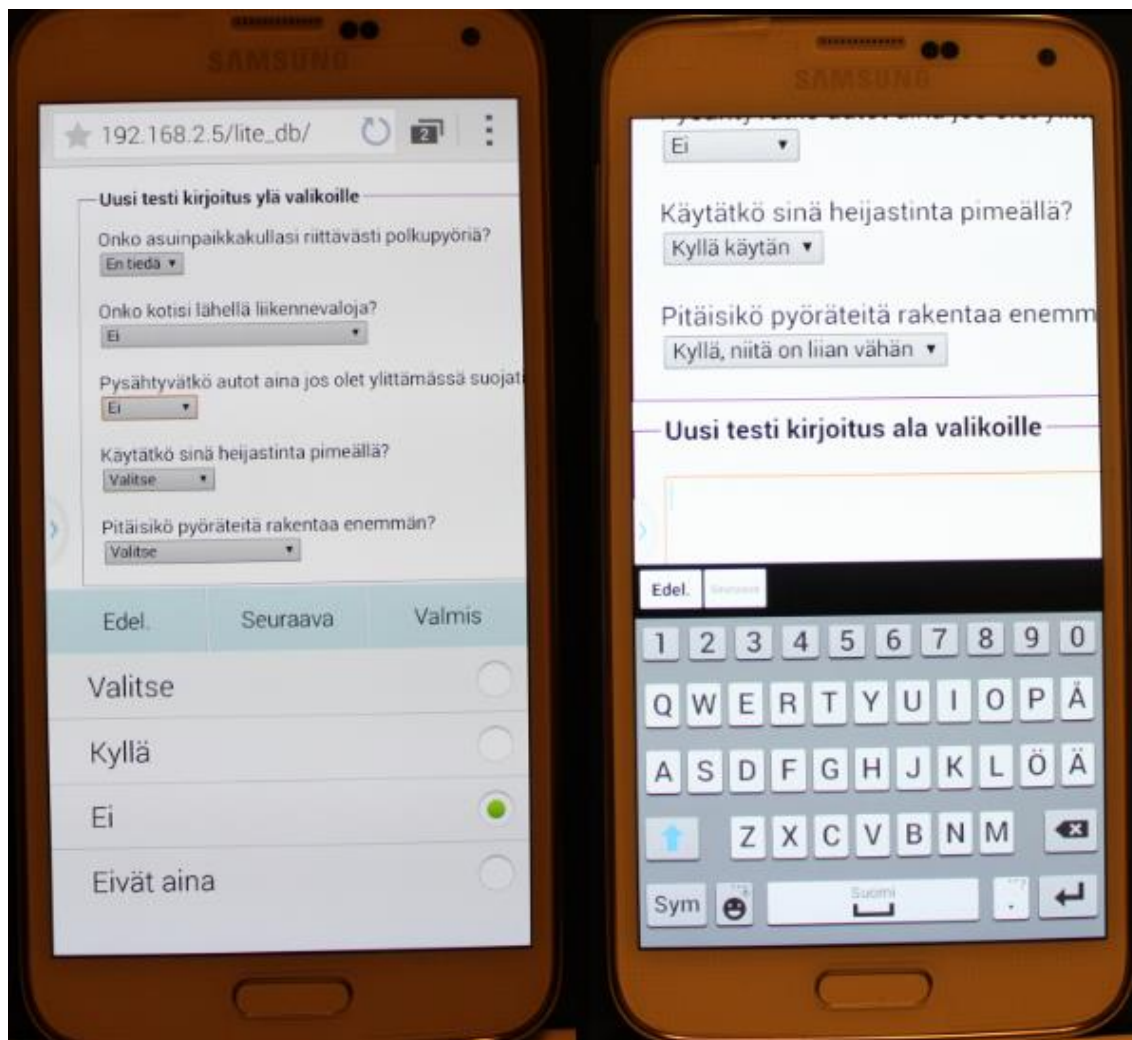
a s d f g h j k l ö ä

↑ z x c v b n m , ! . ? ↑

Sym T Suomi

Kuva 42: Asiakaskyselyyn vastaaminen tablet-laitteella

Kuvan 43 mukaisesti on esillä matkapuhelimella otettu WLAN-yhteys palvelimelle, missä voi samalla tavalla vastaila verkkosovelluksen asiakaskyselyyn. Myös matkapuhelimen valikko muodostuu matkapuhelimen omien asetusten mukaan. Mobiililaitteilla tehtävä asiakaskysely mahdollistaa mobiililaitteiden omien valikkojen käytön, mikä tekee sovelluksesta tutun käyttäjälle.




Kuva 43: Asiakaskyselyyn vastaaminen matkapuhelimella

## 8 Verkkosovelluksen vastauksien vieminen Excel-tiedostoon

Kuvan 44 mukaisesti on nähtävissä, että tietokannassa on paljon vastauksia, vaikka tässä vaiheessa kaikki vastaukset ovat testikierrokselta. Sivustolta näkyy, että valikkojen kysymykset on eroteltu vapaasta tekstikentästä erilleen, jotta sivuston ulkoasu olisi selkeämpi. Lisäksi vastaukset ovat erityyppisiä. Valikkojen vastauksia on mahdollista vierittää vaakatasossa edes takaisin, mikäli vastauksia halutaan tutkia.

Kirjautunut: hallinta



PALAUTE

TAULUKOT

VALIKOT

VIESTIT

TIETOKANTA

Asiakas kysymys 1 - 5 vastaukset

KYSYMYSI	VASTAUS1		KYSYMYSI2	VASTAUS2		KYSYMYSI3	VASTAUS3		KYSYMYSI4	VASTAUS4		KYSYMYSI5	VASTAUS5		KYSYMYSI6	VASTAUS6		KYSYMYSI7	VASTAUS7		KYSYMYSI8	VASTAUS8		KYSYMYSI9	VASTAUS9		KYSYMYSI10	VASTAUS10		KYSYMYSI11	VASTAUS11		KYSYMYSI12	VASTAUS12		KYSYMYSI13	VASTAUS13		KYSYMYSI14	VASTAUS14		KYSYMYSI15	VASTAUS15		KYSYMYSI16	VASTAUS16		KYSYMYSI17	VASTAUS17		KYSYMYSI18	VASTAUS18		KYSYMYSI19	VASTAUS19		KYSYMYSI20	VASTAUS20		KYSYMYSI21	VASTAUS21		KYSYMYSI22	VASTAUS22		KYSYMYSI23	VASTAUS23		KYSYMYSI24	VASTAUS24		KYSYMYSI25	VASTAUS25		KYSYMYSI26	VASTAUS26		KYSYMYSI27	VASTAUS27		KYSYMYSI28	VASTAUS28		KYSYMYSI29	VASTAUS29		KYSYMYSI30	VASTAUS30		KYSYMYSI31	VASTAUS31		KYSYMYSI32	VASTAUS32		KYSYMYSI33	VASTAUS33		KYSYMYSI34	VASTAUS34		KYSYMYSI35	VASTAUS35		KYSYMYSI36	VASTAUS36		KYSYMYSI37	VASTAUS37		KYSYMYSI38	VASTAUS38		KYSYMYSI39	VASTAUS39		KYSYMYSI40	VASTAUS40		KYSYMYSI41	VASTAUS41		KYSYMYSI42	VASTAUS42		KYSYMYSI43	VASTAUS43		KYSYMYSI44	VASTAUS44		KYSYMYSI45	VASTAUS45		KYSYMYSI46	VASTAUS46		KYSYMYSI47	VASTAUS47		KYSYMYSI48	VASTAUS48		KYSYMYSI49	VASTAUS49		KYSYMYSI50	VASTAUS50		KYSYMYSI51	VASTAUS51		KYSYMYSI52	VASTAUS52		KYSYMYSI53	VASTAUS53		KYSYMYSI54	VASTAUS54		KYSYMYSI55	VASTAUS55		KYSYMYSI56	VASTAUS56		KYSYMYSI57	VASTAUS57		KYSYMYSI58	VASTAUS58		KYSYMYSI59	VASTAUS59		KYSYMYSI60	VASTAUS60		KYSYMYSI61	VASTAUS61		KYSYMYSI62	VASTAUS62		KYSYMYSI63	VASTAUS63		KYSYMYSI64	VASTAUS64		KYSYMYSI65	VASTAUS65		KYSYMYSI66	VASTAUS66		KYSYMYSI67	VASTAUS67		KYSYMYSI68	VASTAUS68		KYSYMYSI69	VASTAUS69		KYSYMYSI70	VASTAUS70		KYSYMYSI71	VASTAUS71		KYSYMYSI72	VASTAUS72		KYSYMYSI73	VASTAUS73		KYSYMYSI74	VASTAUS74		KYSYMYSI75	VASTAUS75		KYSYMYSI76	VASTAUS76		KYSYMYSI77	VASTAUS77		KYSYMYSI78	VASTAUS78		KYSYMYSI79	VASTAUS79		KYSYMYSI80	VASTAUS80		KYSYMYSI81	VASTAUS81		KYSYMYSI82	VASTAUS82		KYSYMYSI83	VASTAUS83		KYSYMYSI84	VASTAUS84		KYSYMYSI85	VASTAUS85		KYSYMYSI86	VASTAUS86		KYSYMYSI87	VASTAUS87		KYSYMYSI88	VASTAUS88		KYSYMYSI89	VASTAUS89		KYSYMYSI90	VASTAUS90		KYSYMYSI91	VASTAUS91		KYSYMYSI92	VASTAUS92		KYSYMYSI93	VASTAUS93		KYSYMYSI94	VASTAUS94		KYSYMYSI95	VASTAUS95		KYSYMYSI96	VASTAUS96		KYSYMYSI97	VASTAUS97		KYSYMYSI98	VASTAUS98		KYSYMYSI99	VASTAUS99		KYSYMYSI100	VASTAUS100		KYSYMYSI101	VASTAUS101		KYSYMYSI102	VASTAUS102		KYSYMYSI103	VASTAUS103		KYSYMYSI104	VASTAUS104		KYSYMYSI105	VASTAUS105		KYSYMYSI106	VASTAUS106		KYSYMYSI107	VASTAUS107		KYSYMYSI108	VASTAUS108		KYSYMYSI109	VASTAUS109		KYSYMYSI110	VASTAUS110		KYSYMYSI111	VASTAUS111		KYSYMYSI112	VASTAUS112		KYSYMYSI113	VASTAUS113		KYSYMYSI114	VASTAUS114		KYSYMYSI115	VASTAUS115		KYSYMYSI116	VASTAUS116		KYSYMYSI117	VASTAUS117		KYSYMYSI118	VASTAUS118		KYSYMYSI119	VASTAUS119		KYSYMYSI120	VASTAUS120		KYSYMYSI121	VASTAUS121		KYSYMYSI122	VASTAUS122		KYSYMYSI123	VASTAUS123		KYSYMYSI124	VASTAUS124		KYSYMYSI125	VASTAUS125		KYSYMYSI126	VASTAUS126		KYSYMYSI127	VASTAUS127		KYSYMYSI128	VASTAUS128		KYSYMYSI129	VASTAUS129		KYSYMYSI130	VASTAUS130		KYSYMYSI131	VASTAUS131		KYSYMYSI132	VASTAUS132		KYSYMYSI133	VASTAUS133		KYSYMYSI134	VASTAUS134		KYSYMYSI135	VASTAUS135		KYSYMYSI136	VASTAUS136		KYSYMYSI137	VASTAUS137		KYSYMYSI138	VASTAUS138		KYSYMYSI139	VASTAUS139		KYSYMYSI140	VASTAUS140		KYSYMYSI141	VASTAUS141		KYSYMYSI142	VASTAUS142		KYSYMYSI143	VASTAUS143		KYSYMYSI144	VASTAUS144		KYSYMYSI145	VASTAUS145		KYSYMYSI146	VASTAUS146		KYSYMYSI147	VASTAUS147		KYSYMYSI148	VASTAUS148		KYSYMYSI149	VASTAUS149		KYSYMYSI150	VASTAUS150		KYSYMYSI151	VASTAUS151		KYSYMYSI152	VASTAUS152		KYSYMYSI153	VASTAUS153		KYSYMYSI154
----------	----------	--	-----------	----------	--	-----------	----------	--	-----------	----------	--	-----------	----------	--	-----------	----------	--	-----------	----------	--	-----------	----------	--	-----------	----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	------------	-----------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------	------------	--	-------------

Kuva 44: Asiakkaiden vastauksista hallintasivustolla

Kuvan 45 mukaisesti on esillä PHP-koodilla toteutettu funktio, jonka tehtävänä on luoda koko taulukko PHP-koodilla. Taulukko muodostuu HTML-koodin "fieldsetin"-sisälle. Lisäksi "require"-käskyllä luetaan sivustolta puuttuvat toiminnot.

```
<div id="as_1-5">
<fieldset>
<legend>Asiakas kysymys 1 - 5 vastaukset</legend>
<?php
    require "../selain_update.php";
    require "taulukot_funktiot.php";
    $taulukot_uusi = taulukot_uusi();
    echo $taulukot_uusi;
?>
</fieldset>
</div>
```

Kuva 45: PHP-funktiolla muodostettu taulukko

Kuvassa 46 on näkymä taulukon hallintasivustolta, mihin kerätään kaikki vapaaseen tekstikentän asiakkaan kirjoittamat vastaukset. Taulukon tarkoituksena on antaa asiakkaille mahdollisuus kertoa vapaasti uusista palveluideoista.

**Asiakas kysymys 6 vastaukset**

ID	VASTAUS6
1	Internet WEB kauppaa
2	Sähköposti mainontaa
3	Haluaisin enemmän kassoja
4	Ajanvaraus järjestelmää
5	
6	Haluaisin tänne lisää istumapaikkoja

Kuva 46: Vapaan tekstikentän vastaukset taulukossa

Kuvassa 47 on näkymä hallintasivuilta, josta on mahdollista tulostaa Excel-tiedostoon, taulukot 1 - 6 tai sitten kaikki kuusi taulukkoa kerralla. Tietojen keräämisen apuna on käytetty PHP-funktiota. Tietojen vieminen tapahtuu tulosta-painiketta painamalla.

**Tulostus valikko**

Tulostuksen vaihtoehdot:

Valikoi vaihtoehto

- Valikoi vaihtoehto
- Tulosta Excel tiedostoon vain vastaus 1
- Tulosta Excel tiedostoon vain vastaus 2
- Tulosta Excel tiedostoon vain vastaus 3
- Tulosta Excel tiedostoon vain vastaus 4
- Tulosta Excel tiedostoon vain vastaus 5
- Tulosta Excel tiedostoon vain vastaus 6
- Tulosta Excel tiedostoon kaikki vastaukset**

Kuva 47: Alasvetovalikko tietokannasta tulostus Excel-tiedostoon

Kuvassa 48 on näkymä hallintasivuilta, kun alasvetovalikkosta valitaan jokin vastausvaihtoehtoista. Tulostus on toteutettu PHP:n "if"-käskyillä. Kun "if"-käsky toteutuu, niin koodissa kutsutaan PHP-funktiota, jolla kerätään tietokannasta tiedot Excel-tiedostoon.

```
<?php
if(isset($_POST['tulosta'])) {
    $VALIKKO = $_POST['VALIKKO'];

    if ($VALIKKO == "VASTAUS1")
    {
        $Excel1 = Excel1();
    }
    if ($VALIKKO == "VASTAUS2")
    {
        $Excel2 = Excel2();
    }
    if ($VALIKKO == "VASTAUS3")
    {
        $Excel3 = Excel3();
    }
    if ($VALIKKO == "VASTAUS4")
    {
        $Excel4 = Excel4();
    }
    if ($VALIKKO == "VASTAUS5")
    {
        $Excel5 = Excel5();
    }
    if ($VALIKKO == "VASTAUS6")
    {
        $Excel6 = Excel6();
    }
    if ($VALIKKO == "VASTAUS7")
    {
        $Excel7 = Excel7();
    }
}
?>
```

Kuva 48: PHP-koodi Excel-valikosta

Kuvan 49 mukainen Excel-raportti on sovelluksen loppuraportti, missä vastaukset ovat haettu tietokannasta Excel-tiedostoon. Sovellus laskee automaattisesti sen, kuinka monta samaa tulosta löytyy ja vastauksien määrän sekä vastausprosenttijakauman. Tekstikentän tuloksia voidaan verrata valikkojen tuloksiin ja etsiä yhtäläisyyksiä ja eroavaisuuksia.

ID	KYSYMYS1	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTIA %
	4 Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ammattitaitoista	3		38
	6 Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Epäasiallista	1		13
	8 Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ystävällistä	4		50
				8	
ID	KYSYMYS2	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTIA %
	8 Asiakkaan haluamia tuotteita?	on saatavilla, hinnat merkitty oikein ja täsmällisesti	6		75
	6 Asiakkaan haluamia tuotteita?	on saatavilla, mutta hinnat merkitty väärin	2		25
				8	
ID	KYSYMYS3	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTIA %
	7 Yrityksen luotettavuuden vaikuttaa?	Asiantunteva henkilöstö	2		25
	6 Yrityksen luotettavuuden vaikuttaa?	Henkilökohtaiset kontaktit	2		25
	8 Yrityksen luotettavuuden vaikuttaa?	Laadukkaat tuotteet	3		38
	2 Yrityksen luotettavuuden vaikuttaa?	Maine	1		13
				8	
ID	KYSYMYS4	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTIA %
	6 Ongelmatilanteissa yrityksen työntekijällä?	Ei ole riittävästi vapautta toimia asiakastilanteen mukaan	3		38
	8 Ongelmatilanteissa yrityksen työntekijällä?	On riittävästi vapautta toimia asiakastilanteen mukaan	5		63
				8	
ID	KYSYMYS5	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTIA %
	8 Tullessani yrityksen tiloihin haluaisin huomauttaa?	Ei huomautettavaa	3		38
	3 Tullessani yrityksen tiloihin haluaisin huomauttaa?	Liian kalliista hinnoista	1		13
	6 Tullessani yrityksen tiloihin haluaisin huomauttaa?	Tilojen viihtyvydestä	1		13
	5 Tullessani yrityksen tiloihin haluaisin huomauttaa?	Tuotteiden esillepanosta	3		38
				8	
ID	VAPAA VASTAAMINEN				
	1 Internet WEB kauppa				
	2 Sähköposti mainontaa				
	3 Haluaisin enemmän kassoja				
	4 Ajanvaraus järjestelmää				
	5				
	6 Haluaisin tänne lisää istumapaikkoja				
	7 Internet sivuja olisi hyvä muuttaa				
	8 Haluaisin kattavampia tarjouksia sähköpostiin				

Kuva 49: Tulokset Excel-tiedostossa

## 9 Sivuille kirjautumisen hallinta

Asiakassivustolla on mahdollisuus kirjautua hallintasivuille käyttämällä käyttäjätunnusta ja salasanaa. Käyttäjätunnus ja salasana on kirjoitettu PHP-sivun koodiin. Tämä salaustekniikka on riittävä tämän kaltaiselle ohjelmalle. Md5 on salaustekniikka, jossa toimii 128-bittinen luku ja se esitetään tässä heksadesimaalimuodossa. (Heinisuo 2004, 148.) Tässä PHP-koodissa kuvan 50 mukaisesti käyttäjänimi kirjautumiselle on ”hallinta” ja salasana on ”salalomake”. Md5 kääntää tämän ”salalomake” muotoon ”62dd01b9d708bb9bdb5481e165cf00bc”



Oletuksena ilman kirjautumista "asiakas"-käyttäjä on kirjautuneena. TRACE-tietokanta kertoo sen, kumpi käyttäjistä on kirjaantuneena järjestelmään. Tietokannassa on arvo "on" sen käyttäjän kohdalla kumpi käyttäjistä on aktiivisena. Kun kirjaututaan ja käyttäjätunnus on oikea, niin asetetaan TRACE-tietokantaan arvo "on", jolloin selaimessa "asiakas" nimi vaihtuu "hallinta" nimeksi. Lisäksi hallintasivujen linkit "taulukot", "valikot", "viestit", ja "tietokanta" aktivoituvat. Kun kirjautuneen on asiakaskäyttäjä, niin esillä on ainoastaan linkki "palautte", mikä on myös etusivu.

```
if(isset($_POST['kirjaudu']))
{
    $salasana=$_POST['salasana'];
    $tunnus=$_POST['tunnus'];
    $Salattu=md5($salasana);

    $db = new SQLite3('tietokanta/talo.db');
    $results = $db->query('SELECT * from KAYTTAJAT');
    while ($row = $results->fetchArray())
    {
        if ($row['KAYTTAJA'] == $tunnus)
        {
            ...
            if ($row['SALASANA'] == $Salattu)
            {
                $db->exec("UPDATE KAYTTAJAT SET TRACE='on' WHERE ID='2'");
                $db->exec("UPDATE KAYTTAJAT SET TRACE='ei' WHERE ID='1'");
            }
        }
    }

    $Selain_paivitys = selain_update();
    echo $Selain_paivitys;
}
```

Kuva 50: PHP:n MD5-salauksesta

Kuvassa 51 on näkymä asiakaskäyttäjä on sivustolta. Jos käyttäjänä on kirjautuneen "hallinta"-käyttäjä, niin käyttäjänimi ja salasana poistuvat näytöltä. "Asiakas"-käyttäjältä on piilotettu hallintasivustojen linkit.

Kuva 51: Asiakaskäyttäjän näkymä sivustosta

Kuvassa 52 on näkymä siitä, kun ”hallinta”-käyttäjä on kirjautuneena järjestelmään. ”Hallinta”-käyttäjän uloskirjautuminen on myös palautesivustolla. Sivuston oikeassa ylälaudassa on nähtävillä hallintakäyttäjän nimi kirjautuneena.



Kuva 52: Hallintakäyttäjän näkymä sivustosta

Kuvassa 53 on näkymä siitä, kuinka valikot on piilotettu asiakaskäyttäjältä PHP-koodissa. Koodissa haetaan ”kayttajat”-tietokannasta arvoa ”on”. Jos arvo löytyy, niin sen käyttäjän tiedot näytetään selaimessa.

```
<?php
$db = new SQLite3('tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
    if ($row['KAYTTAJA'] == "asiakas")
    {
        echo '<ul id="linkit">';
        echo '<li id="palaute"><a href="index.php" title="Palaute">PALAUTE</a></li>';
        echo '</ul>';
    }
    else
    {
        echo '<ul id="linkit">';
        echo '<li id="palaute"><a href="index.php" title="Palaute">PALAUTE</a></li>';
        echo '<li id="taulukot"><a href="hallinta/taulukot.php" title="Taulukot">TAULUKOT</a></li>';
        echo '<li id="valikot"><a href="hallinta/valikot.php" title="Ruksit">VALIKOT</a></li>';
        echo '<li id="viestit"><a href="hallinta/viestit.php" title="Viestit">VIESTIT</a></li>';
        echo '<li id="tietokanta"><a href="hallinta/tietokanta.php" title="Tietokanta">TIETOKANTA</a></li>';
        echo '</ul>';
    }
}
?>
```

Kuva 53: PHP-koodi linkkien piilottamisesta asiakaskäyttäjältä

## 9.1 Salasanan vaihtaminen hallintasivustolla

Hallintakäyttäjä pysyy vaihtaa salasanan, mutta ei lisäämään uusia käyttäjiä. Kaksi käyttäjää on riittävä määrä tämän ohjelman toimimiselle. Kuvan 54 mukaisesti on nähtävillä kohta ”vaihda salasana”, missä salasanan vaihto tapahtuu. Mikäli salasanat eivät ole yhtenäiset, niin ohjelma antaa siitä virheilmoituksen käyttäjälle.

Kuva 54: Hallintakäyttäjän salasanan vaihto tietokanta sivustolla

Kuvassa 55 on näkymä siitä, kuinka kyseinen salasanan vaihto tapahtuu PHP-koodissa. Salasana muutetaan edelleen Md5-salauksella. Mikäli salasana ei täsmää, niin siitä tulee virheilmoitus. Myöskään kentät eivät voi olla tyhjiä. Kaikki ehdot pitää täyttyä, ennen kuin uusi salasana voidaan tallettaa järjestelmään.

```
<?php
$seka = $_POST['salasana'];
$stoka = $_POST['salasana2'];

if(isset($_POST['vaihda']))
{
    if (empty($seka) || empty($stoka))
    {
        echo 'Kenttä ei voi olla tyhjiä';
    }
    else
    {
        if ($seka == $stoka)
        {
            $salasana = $seka;
            $salattu_salasana = md5($salasana);

            $db = new SQLite3('../tietokanta/talo.db');
            $db->exec("UPDATE KAYTTAJAT SET SALASANA='$salattu_salasana' WHERE ID=2");
            $db->close();
            echo 'Salasana vaihdettu';
        }
        else
        {
            echo 'Salasanat eivät täsmää';
        }
    }

    $selain_update = selain_update();
    echo $selain_update;
}
?>
```

Kuva 55: Salasanan vaihto PHP-koodina

## 9.2 Tietokantojen poisto

Kuvassa 56 on näkymä siitä, kuinka hallintakäyttäjällä on mahdollisuus poistaa vanhat tietokannat. Tietokannoista poistetaan kaikki muut tietokannat, paitsi asiakkaille tarkoitettujen valikkojen vastaustietokanta. Tämä mahdollistaa sen, että vanhat vastaukset ovat saatavissa, kun tehdään uusi asiakaskysely. Tämä tekee ohjelman toiminnasta yhdenmukaisen edellisen asiakaskyselyn kanssa.

Kuva 56: Salasanan vaihto ja tietokannan poisto

Kuvan 57 mukainen näkymä PHP-koodista on nähtävillä kohta taulukon poistosta. Ohjelma poistaa kaikki muut tietokannat, paitsi sen valikon, mihin tallennetaan talteen kysymysten 1 - 5 vastausvalikot. Joka kerta ei tarvitse tehdä vastauksia alusta asti, jos ne löytyvät valmiiksi tietokannasta edellisen kyselyn jäljiltä.

```
$db = new SQLite3('../tietokanta/valikko2.db');
$db->exec("DROP TABLE IF EXISTS VALIKKO2");
if(!$db)
{
    echo $db->lastErrorMsg();
}
$db->close();
```

Kuva 57: Tietokannan poisto PHP-koodilla

## 10 Yhteenveto ja johtopäätökset

Yhteenvetona voisin sanoa, että ohjelman kokonaisuus toimii hyvin tiedonkeräämisestä loppu-raporttiin saakka. Lisäksi ohjelman kysymykset ovat helposti muutettavissa myös toisenlaiseen käyttöön yrityksen sisäisten asioiden kehittämiseksi. Ohjelma on siis monikäyttöinen eikä pal-jasta, kuka on minkäkin viestin lähettänyt. Ohjelmiston käyttö auttaa vähentämään loppura-portin käsin tehtävää työskentelyä automaattisen laskennan avulla.

Johtopäätöksenä sanoisin, että tällä ohjelmistolla on mahdollisuus aloittaa palveluidenkehittäminen, kunhan on riittävästi asiakasvastauksia. Hyvin paljon on myös siitä kiinni kuinka asiakkaalle esitetyt kysymykset muotoillaan. Myös tämän takia ohjelmiston tekovaiheessa oli painotettu otsikoiden merkitystä. PHP-ohjelmointikielellä on mahdollista toteuttaa todella monimutkaisia sovelluksia. Toisenlaisella tietokannalla, kuten MySQL-tietokanta olisi lähes rajattomat mahdollisuudet tehdä erilaisia asioita. SQLITE3-tietokanta oli tähän projektiin kuitenkin sopiva ratkaisu. Tablet laitteiden ja matkapuhelimen käyttö toimii hyvin ohjelmiston kanssa. Ympäristön kannalta näkisin mieluisesti erilaisten yritysten käytössä tablet laitteistoja paperilomakkeiden sijaan ympäristön kuormittamisen vuoksi.

Opinnäytetyön aikana on ehtinyt tutustumaan monenlaisiin asioihin, niin PHP-ohjelmointikielen kuin SQLITE3-tietokannan kanssa. Uskon, että voin käyttää oppimaani hyväksi myös muiden ohjelmointikielten kanssa. Projektin alusta moni asia oli ennestään tuttua, mutta viimeistään nyt projektin loputtua tietämys eri osa-alueilla on vain lisääntynyt.

Kehitysehdotuksena voisi olla erillinen tietokantojen poisto. Lisäksi käyttäjähallinnan voisi rakentaa eri tavalla käyttäen sessioita niin, että ohjelman ottaa käyttöön Internetissä ja käyttää vahvempaa salasanatekniikkaa. Lisäksi asiakaskysymyksiin voisi lisätä mahdollisuuden käyttää kysymyksissä valokuvaa. Lisäksi automaattinen tietokantojen kopiointi lisäisi käytettävyyttä ongelmatilanteissa.

## Lähteet

Andy, B., Cameron, M & Simon, C. 2006. Viimeisimmät CSS-niksit ja tekniikat. Jyväskylä: Gummerus Kirjapaino Oy.

Heinisuo, R. 2004. PHP ja MySQL tietokantapohjaiset verkkopalvelut. Jyväskylä: Gummerus Kirjapaino Oy.

Hernandez, M. 2000 tietokannan suunnittelu ja toteutus. Jyväskylä: Gummerus kirjapaino Oy

Hiltunen, A. 2012 Johtamisen taito elämänmittainen matka. Sanoma Pro Oy

Hirsijärvi, S. & Remes, P & Sajavaara, P. 2007. Tutki ja kirjoita. Keuruu: Otavan Kirjapaino Oy.

Lämsä, A., & Uusitalo, O. 2009. Palveluiden markkinointi esimiestyön haasteena., 1-6 painos. Helsinki: Edita Prima Oy.

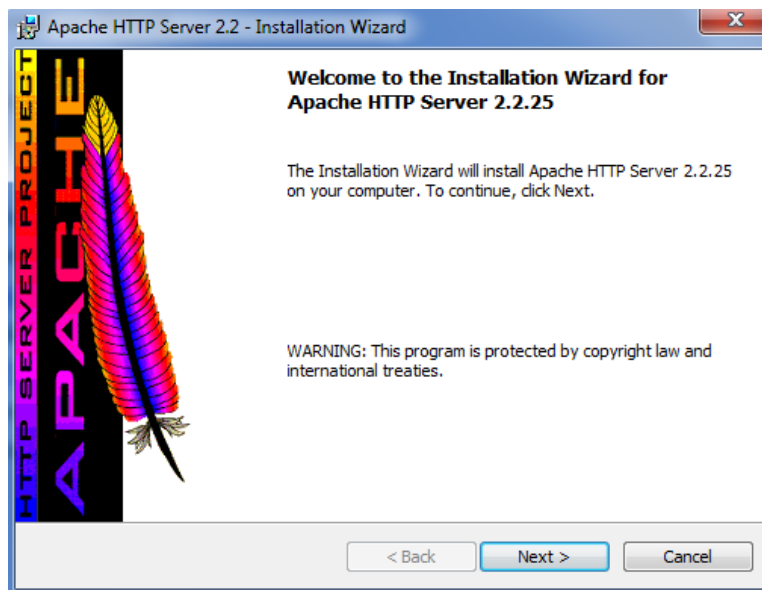
Sinkkonen, I., Kuoppala, H., Parkkinen, J. & Vastamäki, R. 2006. Käytettävyyden psykologia. Helsinki: Edita Prima Oy.

Vilkka, H. 2005. Tutki ja kehitä. Keuruu: Otavan kirjapaino Oy.

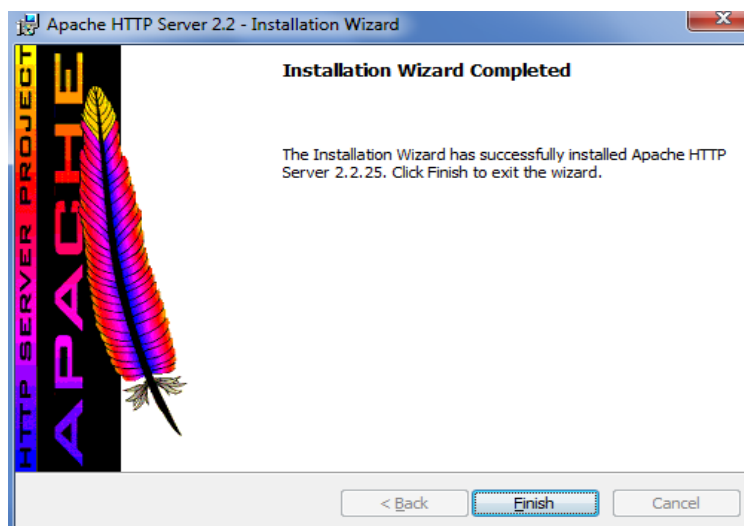
Vilkka, H. 2006. Tutki ja kehitä. Vaajakoski: Gummerus kirjapaino Oy.

Wiio, A. 2004. Käyttäjäystävällisen sovelluksen suunnittelu. Helsinki: Edita Prima Oy.

Kuvat



Kuva 1: Apache palvelimen asennuksen aloituskuva



Kuva 2: Installation wizard Completed sivu

```
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so

# -----PHP MODULLI -----
LoadModule php5_module "c:/php/php5apache2_2.dll"
AddHandler application/x-httpd-php .php

# configure the path to php.ini
PHPIniDir "c:/php"
```

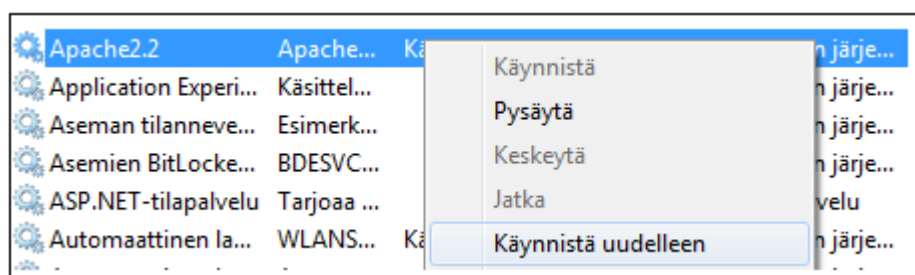
Kuva 3: PHP:n aktivointi httpd.conf tiedostossa

```
; Directory in which the loadable extensions (modules) reside.
; http://php.net/extension-dir
; extension_dir = "."
; On windows:
extension_dir = "c:\php\ext"
```

Kuva 4: Extension\_dir asetuksen muuttaminen

```
; Windows Extensions
;
;extension=php_bz2.dll
;extension=php_curl.dll
extension=php_sqlite3.dll
```

Kuva 5: Windows-Extensions asetuksen muuttaminen




Kuva 6: Apache2.2 uudelleen käynnistys Windows palveluista



Kuva 7: Phpinfo-ohjelman logo



CSS koodi	Sivuston reunukset kuvana	HTML koodi
<pre>#wrapper { width: 800px; } #kirjautuminen { width: 800px; } #Logo { clear: both; } .linkki_laatikko { width: 800px; height: 35px; } fieldset { }</pre>	 <p>The diagram illustrates the visual structure of the page layout. It consists of four main rectangular sections stacked vertically. The top three sections are thin horizontal bars: the first is labeled 'kirjautuminen', the second 'Logo', and the third 'linkki_laatikko'. Below these is a large, empty rectangular box labeled 'fieldset'.</p>	<pre>&lt;div id="wrapper"&gt; &lt;div id="kirjautuminen"&gt; &lt;/div&gt;  &lt;div id="Logo"&gt; &lt;img src="../../kuvat/logo.png"&gt; &lt;/div&gt;  &lt;div class="linkki_laatikko"&gt; &lt;/div&gt;  &lt;fieldset&gt; HTML sisältö tähän &lt;/fieldset&gt; &lt;/div&gt; &lt;!-- &lt;div id="wrapper"&gt; loppuu--&gt;</pre>

Kuva 8: CSS- ja HTML-koodi käyttöliittymän reunukset kuvana

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

*Profiili Palvelut Oy*

**PALAUTE**

**Oletustekstiä ei ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.**

**Oletustekstiä ei ole muutettu hallintasivuilla.**

Asiakkaan kysymys 1 puuttuu

Asiakkaan kysymys 2 puuttuu

Asiakkaan kysymys 3 puuttuu

Asiakkaan kysymys 4 puuttuu

Asiakkaan kysymys 5 puuttuu

Kuva 9: Oletusnäkö sivustolta ilman asetuksia

**Asiakas kysymys 1**

Asiakkaan kysymys 1 puuttuu

Tallenna

Kuva 10: Asiakaskysymyksen tallentaminen

```
<textarea name="as_kys1_textbox" cols="15" rows="3">
<?php
    $Asiakaskysymys1 = Asiakaskysymys1();
    echo $Asiakaskysymys1;
?>
</textarea>
```

Kuva 11: HTML- ja PHP-koodi

```
function Asiakaskysymys1 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=1');
echo $row;
}
```

Kuva 12: Asiakaskysymys 1 PHP-funktion sisältö

**Asiakas kysymys 1**

Toimiikohan tämä  
kysymys vai ei?

Kuva 13: Hallintasivustolla asetetaan kysymys asiakkaalle

```
<?php
    $as_kys1_textbox = $_POST['as_kys1_textbox'];
    if(isset($_POST['nappula1_kys']))
    {
        $db = new SQLite3('../tietokanta/kysymykset.db');
        $db->exec("UPDATE KYSYMYKSET SET KYSYMYKSET='$as_kys1_textbox' WHERE ID=1");
        $db->close();

        $selain_update = selain_update();
        echo $selain_update;
    }
?>
```

Kuva 14: PHP-koodilla päivitetään kysymys tietokantaan

ID	KYSYMYS1	KYSYMYS2
1	Toimiikohan tämä kysymys vai ei?	
2		Asiakkaan kysymys 2 puuttuu
3		
4		
5		

Kuva 15: Näkymä tietokannassa olevista tietueista

**Kysymys 1 valikon testaaminen**

Toimiikohan tämä kysymys vai ei?

Kuva 16: Kysymys 1 testaaminen hallintasivustolla

**Kysymys1 vastauksen lisääminen**

Testi vastaus

Tallenna

Kuva 17: Vastauksen lisääminen valikkoon

```
<?php
if (isset($_POST['nappula1_lis']))
{
    $textareal_lis = $_POST['textareal_lis'];
    $db = new SQLite3('../tietokanta/valikko1_lisaa.db');
    $db->exec("INSERT INTO VALIKKO1_LISAA (VALIKKO) VALUES
    ('$textareal_lis')");
    $db->close();

    $selain_update = selain_update();
    echo $selain_update;
}
?>
```

Kuva 18: PHP-koodi valikkoon lisättävästä vastauksesta tietokantaan

ID	VALIKKO
1	Valitse
2	Testi vastaus

Kuva 19: Näkymä SQLITE3-tietokannasta

**Kysymys 1 vastauksien varsto**

ID	VALIKKO	TOIMINNOT
1	Valitse	poista Lisää
5	Ystävällistä	poista Lisää
7	Ammattitaitoista	poista Lisää
8	Epäasiallista	poista Lisää
10	Välinpitämätöntä	poista Lisää
11	Testi vastaus	poista Lisää

**Kysymys 1 valikon järjestys**

ID	VALIKKO	TOIMINNOT
1	Valitse	poista

**Kysymys 1 valikon testaaminen**

Asiakkaan kysymys 1 puuttuu

Valitse ▼

Kuva 20: Alasvetovalikon vastaukset

```

<?php
if (isset($_POST['valikko1_lisaa_poista']))
{
    $valikko1_lisaa_id = $_POST['valikko1_lisaa_id'];
    $db = new SQLite3('../tietokanta/valikko1_lisaa.db');
    $db->exec("DELETE FROM VALIKKO1_LISAA WHERE ID='$valikko1_lisaa_id'");
    $db->close();

    $selain_update = selain_update();
    echo $selain_update;
}

if (isset($_POST['valikko1_lisaa']))
{
    $valikko1_lisaa_id = $_POST['valikko1_lisaa_id'];
    $valikko1_lisaa_kysymys = $_POST['valikko1_lisaa_kysymys'];

    $db = new SQLite3('../tietokanta/valikko1.db');
    $db->exec("INSERT INTO VALIKKO1 (ID,VALIKKO) VALUES
('$valikko1_lisaa_id','$valikko1_lisaa_kysymys')");
    $db->close();

    $selain_update = selain_update();
    echo $selain_update;
}
?>

```

Kuva 21: PHP-koodi lisää- ja poista-painikkeista

Toimiikohan tämä kysymys vai ei?

Valitse ▼

Valitse

Testi vastaus

Kuva 22: Alasvetovalikon testaaminen

**Oletustekstiä ei ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.**

Kuva 23: Oletusviesti punaisella taustalla

```
else //arvo1
{
    $db = new SQLite3('tietokanta/viestit.db');
    $row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=1');

    if ($row == "oletus") // Arvo 2
    {
        echo '<p> <span class="virhe">Oletustekstiä ei
ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.</span></p>';
    }
    else // Arvo 2
    {
        $Tervehdys = Asiakkaan_alkutervehdys();
        echo $Tervehdys;
    } // Arvo 2
} // arvo1
$db->close();
?>
```

Kuva 24: Oletusviesti punaisella taustalla PHP-koodina

**Alkutervehdys**

Alkutervehdys  
asiakkaalle tähän

Tallenna

Kuva 25: Alkutervehdyksen muuttaminen

```
<textarea name="as_kys1_textbox" cols="15" rows="4"><?php
$Alkuteksti = Alkuteksti();
echo $Alkuteksti;
?></textarea>
```

Kuva 26: PHP-funktiolla luetaan tiedot tekstikentän sisälle

```
function Alkuteksti ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=1');
echo $row;
}
```

Kuva 27: PHP-funktion luku VIESTI-tietokannasta

```
<?php
$as_kys1_textbox = $_POST['as_kys1_textbox'];
if(isset($_POST['nappula1_kys']))
{
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET VIESTI='$as_kys1_textbox' WHERE ID=1");
$db->close();

$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=1");
$db->close();

$selain_update = selain_update();
echo $selain_update;
}
?>
```

Kuva 28: Alkuterveydyksen lukeminen PHP-koodilla tietokannasta

**Alkuterveyhdys**

Muutetaan  
alkuterveyhdys  
asiakkaalle

Tallenna

Kuva 29: Alkuterveydyksen muuttaminen hallintasivustolla

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Palute viesti asiakkaalle tähän	oletus
3	Ylävalikon selitys asiakkaalle tähän	oletus
4	Alavalikon selitys asiakkaalle tähän	oletus
5	tyhjä	ei

Kuva 30: VIESTIT-tietokannan muutokset

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

*Profiili Palvelut Oy*

**PALAUTE**

**Muutetaan alkutervehdys asiakkaalle**

Oletustekstiä ei ole muutettu hallintasivuilla.

Toimiikohan tämä kysymys vai ei?

Valitse

Kuva 31: Uusi alkutervehdys asiakkaalle

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

*Profiili Palvelut Oy*

**PALAUTE**

**Oletustekstiä ei ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.**

Oletustekstiä ei ole muutettu hallintasivuilla.

Kuva 32: Kiitos palautteesta punaisella taustalla



**Palute viestistä**

Kiitos testi  
 palautteesta

Tallenna

Kuva 33: Viestin tallentaminen tietokantaan

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Kiitos testi palautteesta	ei
3	Ylävalikon selitys asiakkaalle tähän	oletus
4	Alavalikon selitys asiakkaalle tähän	oletus
5	tyhjä	ei

Kuva 34: Palaute tallennettu tietokantaan

Käyttäjänimi:  Salasana:  Kirjaudu Kirjautunut: asiakas

**Profili Palvelut Oy**

**PALAUTE**

**Kiitos testi palautteesta**

Oletus tekstiä ei ole muutettu hallinta sivuilla.

Toimiikohan tämä kysymys vai ei?  
 Valitse

Kuva 35: Ilmoitus ilman punaista taustaväriä

**Ylävalikon teksti**

Uusi testi  
 kirjoitus ylä  
 valikoille

Tallenna

Kuva 36: Ylävalikon tekstin muuttaminen

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Kiitos testi palautteesta	ei
3	Uusi testi kirjoitus ylä valikoille	ei
4	Alavalikon selitys asiakkaalle tähän	oletus
5	tyhjä	ei

Kuva 37: Viesti ylävalikolle tietokannassa

**Oletustekstiä ei ole muutettu hallintasivuilla.**

Tallenna

Kuva 38: Alavalikon oletusteksti punaisella taustavärillä

**Alavalikon teksti**

Uusi testi  
kirjoitus ala  
valikoille

Tallenna

Kuva 39: Alavalikon tekstin muuttaminen

ID	VIESTI	OLETUS
1	Muutetaan alkutervehdys asiakkaalle	ei
2	Kiitos testi palautteesta	ei
3	Uusi testi kirjoitus ylä valikoille	ei
4	Uusi testi kirjoitus ala valikoille	ei
5	tyhjä	ei

Kuva 40: VIESTI-tietokanta muutoksen jälkeen rivillä 4

**Uusi testi kirjoitus ala valikoille**

Tallenna

Kuva 41: Alavalikon oletusteksti on muutettu

PALAUTE

## Palvelutuloa antamaan palautetta

**Palveluvalikko**

Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?

Ammattitaitoista

Asiakkaan haluamia tuotteita?

on saatavilla, hinnat merkitty oikein ja täsmällisesti

Yrityksen luotettavuuten vaikuttaa?

Valitse

Onglematilanteissa yrityksen työntekijällä?

Valitse

Tullessani yrityksen tiloihin haluaisin huomauttaa?

Valitse

**Mitä palvelua sinä muuttaisit paremmaksi?**

Haluaisin lisää ammattikirjallisuutta

Edel.    Seur.

1 2 3 4 5 6 7 8 9 0

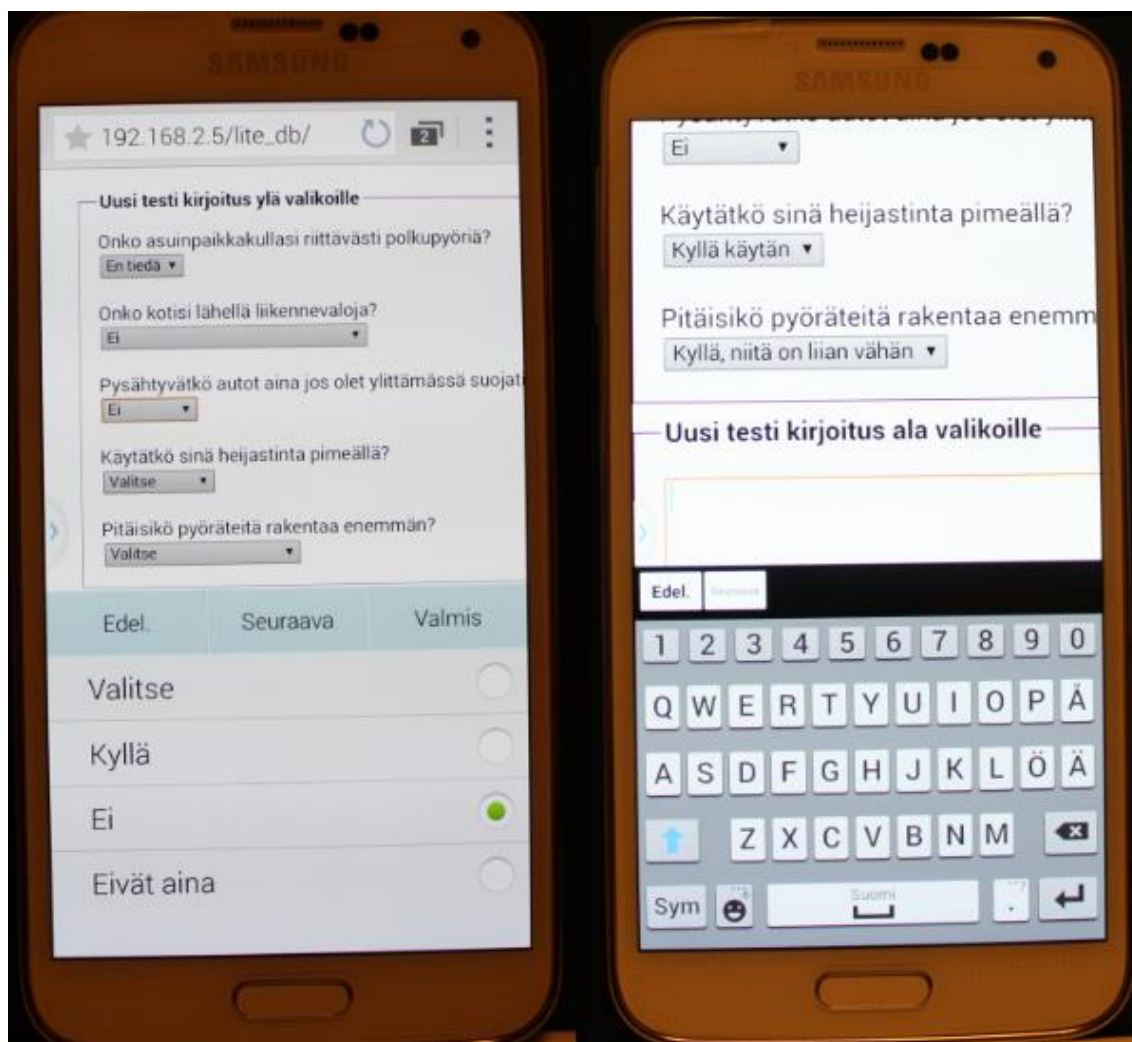
q w e r t y u i o p å

a s d f g h j k l ö ä

↑ z x c v b n m , ! . ? ↑


Sym T Suomi

Kuva 42: Asiakaskyselyyn vastaaminen tablet-laitteella



Kuva 43: Asiakaskyselyyn vastaaminen matkapuhelimella

Kirjautunut: hallinta



PALAUTE
 **TAULUKOT**
 VALIKOT
 VIESTIT
 TIETOKANTA

Asiakas kysymys 1 - 5 vastaukset
 

KYSYMYSI	VASTAUS1	KYSYMYSI2	VASTAUS2	KYSYMYSI3	VASTAUS3	K
Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ammattitaitoista	Asiakkaan haluamia tuotteita?	on saatavilla, hinnat merkitty oikein ja täsmällisesti	Yrityksen luotettavuuden vaikuttaa?	Asiantunteva henkilöstö	Ongle yrityk työnt
Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ystävällistä	Asiakkaan haluamia tuotteita?	on saatavilla, hinnat merkitty oikein ja täsmällisesti	Yrityksen luotettavuuden vaikuttaa?	Maine	Ongle yrityk työnt
Kun asioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ammattitaitoista	Asiakkaan haluamia tuotteita?	on saatavilla, mutta hinnat merkitty väärin	Yrityksen luotettavuuden vaikuttaa?	Henkilökohtaiset kontaktit	Ongle yrityk työnt
Kun asioit ensimmäisen		Asiakkaan	on saatavilla, hinnat	Yrityksen	Asiantunteva	Ongle

Kuva 44: Näkymä asiakkaiden vastauksista hallintasivustolla

```

<div id="as_1-5">
<fieldset>
<legend>Asiakas kysymys 1 - 5 vastaukset</legend>
<?php
    require "../selain_update.php";
    require "taulukot_funktiot.php";
    $taulukot_uusi = taulukot_uusi();
    echo $taulukot_uusi;
?>
</fieldset>
</div>

```

Kuva 45: PHP-koodilla taulukoiden muodostaminen funktio-käskyllä

**Asiakas kysymys 6 vastaukset**

ID	VASTAUS6
1	Internet WEB kauppaa
2	Sähköposti mainontaa
3	Haluaisin enemmän kassoja
4	Ajanvaraus järjestelmää
5	
6	Haluaisin tänne lisää istumapaikkoja

Kuva 46: Tekstikentän vastaukset taulukossa

**Tulostus valikko**

Tulostuksen vaihtoehdot:

Valikoi vaihtoehto

- Valikoi vaihtoehto
- Tulosta Excel tiedostoon vain vastaus 1
- Tulosta Excel tiedostoon vain vastaus 2
- Tulosta Excel tiedostoon vain vastaus 3
- Tulosta Excel tiedostoon vain vastaus 4
- Tulosta Excel tiedostoon vain vastaus 5
- Tulosta Excel tiedostoon vain vastaus 6
- Tulosta Excel tiedostoon kaikki vastaukset

Kuva 47: Tietokannasta tulostus Excel-tiedostoon

```
<?php
if(isset($_POST['tulosta'])){
$VALIKKO = $_POST['VALIKKO'];

    if ($VALIKKO == "VASTAUS1")
    {
$Excel1 = Excel1();
    }
    if ($VALIKKO == "VASTAUS2")
    {
$Excel2 = Excel2();
    }
    if ($VALIKKO == "VASTAUS3")
    {
$Excel3 = Excel3();
    }
    if ($VALIKKO == "VASTAUS4")
    {
$Excel4 = Excel4();
    }
    if ($VALIKKO == "VASTAUS5")
    {
$Excel5 = Excel5();
    }
    if ($VALIKKO == "VASTAUS6")
    {
$Excel6 = Excel6();
    }
    if ($VALIKKO == "VASTAUS7")
    {
$Excel7 = Excel7();
    }
}}
?>
```

Kuva 48: PHP-koodi Excel-tiedostoon tulostamisesta



ID	KYSYMYS	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTI A %
4	Kun esioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ammattitaitolta	3		38
6	Kun esioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Epäasiallista	1		13
8	Kun esioit ensimmäisen kerran asiakaspalvelussa oliko palvelu?	Ystävällistä	4		50
				8	
ID	KYSYMYS	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTI A %
8	Asiakkaan haluamia tuotteita?	onsastavilla, hinnat merkitty oikein ja täsmällisesti	6		75
6	Asiakkaan haluamia tuotteita?	onsastavilla, mutta hinnat merkitty väärin	2		25
				8	
ID	KYSYMYS	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTI A %
7	Yhteyksen luotettavuuden vai kuttaa?	Asiantunteva henkilöstö	2		25
6	Yhteyksen luotettavuuden vai kuttaa?	Henkilökohtaiset kontaktit	2		25
8	Yhteyksen luotettavuuden vai kuttaa?	Laadukkaat tuotteet	3		38
2	Yhteyksen luotettavuuden vai kuttaa?	Maine	1		13
				8	
ID	KYSYMYS	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTI A %
6	Ongelmatilanteissa yrityksen työntekijä?	En ole riittävästi vapautta toimia asiakastilanteen mukaan	3		38
8	Ongelmatilanteissa yrityksen työntekijä?	Ori riittävästi vapautta toimia asiakastilanteen mukaan	5		63
				8	
ID	KYSYMYS	VASTAUKSET	SAMOJA	VASTAUKSIA	PROSENTTI A %
8	Tullessani yrityksen tiloihin halusin huomauttaa?	En huomautettavias	3		38
3	Tullessani yrityksen tiloihin halusin huomauttaa?	Uien kalliista hinnoista	1		13
6	Tullessani yrityksen tiloihin halusin huomauttaa?	Tilojen viihtyvydestä	1		13
3	Tullessani yrityksen tiloihin halusin huomauttaa?	Tuotteiden esillepanosta	3		38
				8	
ID	VAPAA VASTAAMINEN				
1	Internet WEB kauppa				
2	Sähköposti mainonta				
3	Haluaisin enemmän kassoja				
4	Ajanvaraus järjestelmää				
5					
6	Haluaisin tänne lisää istumapaikkoja				
7	Internet sivuja olisi hyvä muuttaa				
8	Haluaisin kattavampia tarjouksia sähköpostiin				

Kuva 49: Tulokset Excel-tiedostossa

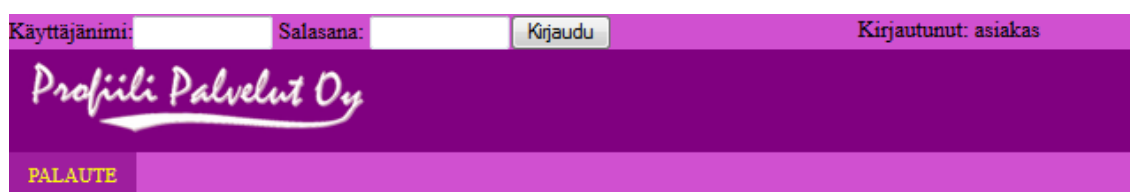
```

if(isset($_POST['kirjaudu']))
{
    $salasana=$_POST['salasana'];
    $tunnus=$_POST['tunnus'];
    $Salattu=md5($salasana);

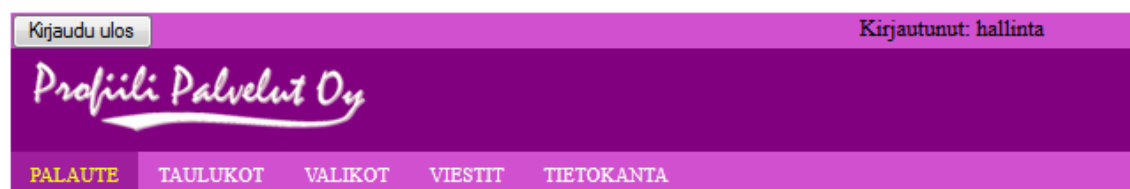
    $db = new SQLite3('tietokanta/talo.db');
    $results = $db->query('SELECT * from KAYTTAJAT');
    while ($row = $results->fetchArray())
    {
        if ($row['KAYTTAJA'] == $tunnus)
        {
            ...
            if ($row['SALASANA'] == $Salattu)
            {
                $db->exec("UPDATE KAYTTAJAT SET TRACE='on' WHERE ID='2'");
                $db->exec("UPDATE KAYTTAJAT SET TRACE='ei' WHERE ID='1'");
            }
        }
    }
    $Selain_paivitys = selain_update();
    echo $Selain_paivitys;
}

```

Kuva 50: PHP-koodi MD5-salauksesta



Kuva 51: Asiakaskäyttäjän näkymä



Kuva 52: Hallintakäyttäjän näkymä

```

<?php
$db = new SQLite3('tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
    if ($row['KAYTTAJA'] == "asiakas")
    {
        echo '<ul id="linkit">';
        echo '<li id="palaute"><a href="index.php" title="Palaute">PALAUTE</a></li>';
        echo '</ul>';
    }
    else
    {
        echo '<ul id="linkit">';
        echo '<li id="palaute"><a href="index.php" title="Palaute">PALAUTE</a></li>';
        echo '<li id="taulukot"><a href="hallinta/taulukot.php" title="Taulukot">TAULUKOT</a></li>';
        echo '<li id="valikot"><a href="hallinta/valikot.php" title="Valikot">VALIKOT</a></li>';
        echo '<li id="viestit"><a href="hallinta/viestit.php" title="Viestit">VIESTIT</a></li>';
        echo '<li id="tietokanta"><a href="hallinta/tietokanta.php" title="Tietokanta">TIETOKANTA</a></li>';
        echo '</ul>';
    }
}
?>

```

Kuva 53: PHP-koodi linkkien piilottamisesta asiakaskäyttäjältä

Kirjautunut: hallinta

*Profili Palvelut Oy*

PALAUTE   TAULUKOT   VALIKOT   VIESTIT   **TIETOKANTA**

**Poista kaikki tietokannat**

Poista taulukot

**Vaihda salasana**

Salasana:

Uudestaan:

Vaihda

Kuva 54: Salasanan vaihto tietokanta sivustolla

```

<?php
$seka = $_POST['salasana'];
$toka = $_POST['salasana2'];

if(isset($_POST['vaihda']))
{
    if (empty($seka) || empty($toka))
    {
        echo'Kenttä ei voi olla tyhjiä';
    }
    else
    {
        if ($seka == $toka)
        {
            $salasana = $seka;
            $Salattu_salasana = md5($salasana);

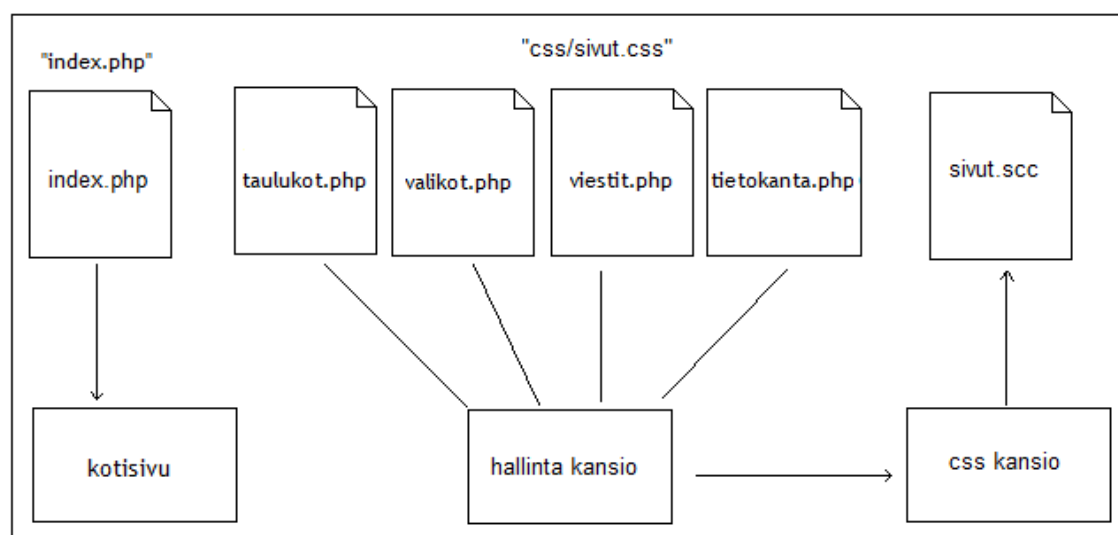
            $db = new SQLite3('../tietokanta/talo.db');
            $db->exec("UPDATE KAYTTAJAT SET SALASANA='$Salattu_salasana' WHERE ID=2");
            $db->close();
            echo'Salasana vaihdettu';
        }
        else
        {
            echo'Salasanat eivät täsmää';
        }
    }

    $selain_update = selain_update();
    echo $selain_update;
}
?>

```

Kuva 55: Salasanan vaihto PHP-koodina

## Kaaviot



Kaavio 1: CSS-tyylitiedostojen sijainnit

## Liitteet

Liite 1	index.php .....	79
Liite 2	db_luonti.php .....	96
Liite 3	Asiakaskysymykset_luku.php .....	101
Liite 4	selain_update.php .....	102
Liite 5	viestit.php .....	103
Liite 6	hallinta/Excel_function.php .....	104
Liite 7	hallinta/Excel_valinnat.php .....	115
Liite 8	hallinta/hallinta_kysymykset_luku.php .....	116
Liite 9	hallinta/poisto.php .....	117
Liite 10	hallinta/taulukot.php .....	119
Liite 11	hallinta/taulukot_funktiot.php .....	122
Liite 12	hallinta/taulukot_poisto.php .....	131
Liite 13	hallinta/tietokanta.php .....	134
Liite 14	hallinta/valikot.php .....	137
Liite 15	hallinta/viesti_kysymykset.php .....	157
Liite 16	hallinta/viestit.php .....	158
Liite 17	css/sivut.css .....	164

Liite 1          index.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>PALAUTE</title>
<style type="text/css">
<!--
```

```
#Asiakas_kysymys {
width: auto;
height: auto;
float: left;
background: white;
text-align: center;
}
```

```
#Asiakas_kysymys h1, h2, p{
padding-right: 10px;
}
```

```
#vastauksen_lisaaminen {
width: auto;
height: auto;
float: left;
background: white;
padding-left: 30px;
```

```
}
#vastauksen_lisaaminen th{
font-size : 12px;
}
```

```
#vastauksen_lisaaminen li{
padding-left: 10px;
```

```
padding-right: 10px;  
}
```

```
#Logo {  
clear: both;  
background: purple;  
}
```

```
#as_vastaukset {  
width: auto;  
height: auto;  
float: left;  
background: white;  
padding-left: 30px;  
}
```

```
#as_vastaukset th{  
font-size : 12px;  
}
```

```
#as_lomake1 {  
width: auto;  
height: auto;  
float: left;  
background: white;  
padding-left: 100px;  
}
```

```
#as_lomake1 th{  
font-size : 12px;  
}
```

```
#as_1-5 {  
width: auto;  
height: auto;  
float: left;  
background: white;  
padding-left: 5px;  
}
```



```
#as_1-5 th{  
    font-size : 12px;  
}
```

```
#sivu5 {  
width: 750px;  
height: 20px;  
float: left;  
background: #D050D0;  
}
```

```
#wrapper {  
width: 800px;  
position: relative;  
left: 15%;  
background: white;  
}
```

```
/* ---.linkki_laatikko div elementti alkaa tästä---*/  
.linkki_laatikko {  
background: #D050D0;  
width: 800px;  
height: 35px;  
}  
/* ---.linkki_laatikko div elementti loppuu tähän---*/
```

```
/* ---#linkit valikon tyyli muotoilu alkaa tästä---*/  
#linkit a:hover, a:active ,  
.palaute #linkit #palaute a,  
.taulukot #linkit #taulukot a,  
.valikot #linkit #valikot a,  
.viestit #linkit #viestit a,  
.poisto #linkit #poisto a {  
background: #9E1E9E; color: yellow;  
}
```

```
#linkit a:link  
.palaute #linkit #palaute a,  
.taulukot #linkit #taulukot a,
```

```
.valikot #linkit #valikot a,  
.viestit #linkit #viestit a,  
.poisto #linkit #poisto a {  
background: #9E1E9E; color: yellow;  
}
```

```
ul {  
margin: 0;  
padding: 0;  
float: left;  
width: 800px;  
height: 20px;  
list-style: none;  
text-transform: uppercase;  
}
```

```
ul li {  
float: left;  
}
```

```
ul a {  
padding: 0 1em;  
line-height: 35px;  
text-decoration: none;  
color: white;  
float: left;
```

```
font-size: 14px;
```

```
}
```

```
fieldset {  
margin: 0.5em 0; /* tekee välin yläreunaan*/  
padding: 1em; /* Sisältötekstin sijoittaminen lomakkeen laitaan*/  
border : 1px solid purple; /* Reunaviivan paksuus ja sen väri*/  
}
```

```

legend {
    font-weight: bold;
}

.virhe {
    background-color: red;
}

#sivu5 {
    width: 750px;
    height: 20px;
    float: left;
    background: #D050D0;
}

#kirjautuminen {
    width: 800px;
    height: 25px;
    float: left;
    background: #D050D0;
}

#oikea{
    width: 200px;
    height: 25px;
    float: right;
    background: #D050D0;
}
-->
</style>
</head>

<body>
<?php
if (!file_exists("tietokanta"))
{mkdir("tietokanta", NULL);}
?>

<?php require "db_luonti.php";?>

```

```
<div id="wrapper">

<div id="kirjautuminen">
<form method="post" action="index.php">
<?php
$db = new SQLite3('tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
if ($row['KAYTTAJA'] == "asiakas")
{
echo '<label for="tunnus">Käyttäjänimi:</label>';
echo '<input type="text" size="12" name="tunnus"/>';
echo '<label for="salasana">Salasana:</label>';
echo '<input type="password" size="12" name="salasana"/>';
echo '<INPUT TYPE="SUBMIT" VALUE="Kirjaudu" NAME="kirjaudu">';
}
else
{
echo '<INPUT TYPE="SUBMIT" VALUE="Kirjaudu ulos" NAME="ulos">';
}
}
?>

<div id="oikea">
<?php
$db = new SQLite3('tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
if ($row['TRACE'] == "on")
{
echo '<label for="salasana">Kirjautunut: </label>';
echo "<td>" . $row['KAYTTAJA'] . "</td>";
}
}
?>
```

```

</div>
</form>
</div>

<div id="Logo">

</div>

<div class="linkki_laatikko">
<div class="palaute">
<?php
$db = new SQLite3('tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
    if ($row['KAYTTAJA'] == "asiakas")
    {
        echo '<ul id="linkit">';
        echo '<li id="palaute"><a href="index.php" title="Palaute">PALAUTE</a></li>';
        echo '</ul>';
    }
    else
    {
        echo '<ul id="linkit">';
        echo '<li id="palaute"><a href="index.php" title="Palaute">PALAUTE</a></li>';
        echo '<li id="taulukot"><a href="hallinta/taulukot.php" tit-
le="Taulukot">TAULUKOT</a></li>';
        echo '<li id="valikot"><a href="hallinta/valikot.php" tit-
le="Ruksit">VALIKOT</a></li>';
        echo '<li id="viestit"><a href="hallinta/viestit.php" tit-
le="Viestit">VIESTIT</a></li>';
        echo '<li id="tietokanta"><a href="hallinta/tietokanta.php" tit-
le="Tietokanta">TIETOKANTA</a></li>';
        echo '</ul>';
    }
}
?>

</div>

```

```

<!-- <div class="ruksit"> LOPPUU TÄHÄN-->

</div>
<!-- <div class="linkki_laatikko"> LOPPUU TÄHÄN-->
<fieldset>
  <?php
require "Asiakaskysymykset_luku.php";
require "selain_update.php";
?>

<?php

if (file_exists("tietokanta/PROFIILI_Oy.db"))
{
else
{
                require "db_luonti.php";
}

if (file_exists("tietokanta/viestit.db"))
{

else
{
                require "db_luonti.php";
}
?>
<h2>

<?php require "viestit.php";?>
<?php
// Tarkistetaan, että onko $row == "on"
                $db = new SQLite3('tietokanta/viestit.db');
                $row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=5');

if ($row == "on") //arvo1
{

                $db = new SQLite3('tietokanta/viestit.db');

```

```
$row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=2');

if ($row == "oletus") // Arvo 4
{
echo '<p> <span class="virhe">Kiitos palautteesta tekstiä ei
ole muutettu hallintasivuilla. </span></p>';

$Muuta_arvo_ei_tilaan = VIESTI_TOTTA_ei();
echo $Muuta_arvo_ei_tilaan;

$Selain_paivitys = selain_update();
echo $Selain_paivitys;
}
else // Arvo 4
{

$Palaute = Palaute_asiakkaalle();
echo $Palaute;

$Muuta_arvo_ei_tilaan = VIESTI_TOTTA_ei();
echo $Muuta_arvo_ei_tilaan;

$Selain_paivitys = selain_update();
echo $Selain_paivitys;
}          // Arvo 4

}          //arvo1

else //arvo1
{

$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=1');

if ($row == "oletus") // Arvo 2
{

echo '<p> <span class="virhe">Oletustekstiä ei
ole muutettu hallintasivuilla. Ole hyvä ja muuta teksti, kiitos.</span></p>';
```

```
    }
    else // Arvo 2
    {

        $Tervehdys = Asiakkaan_alkutervehdys();
        echo $Tervehdys;
    } // Arvo 2

} // arvo1

$db->close();

?>
</h2>
<!-- Asiakas kysymykset 1 HTML asetukset alkaa tästä-->

<div id="as_lomake1">
<fieldset>
<legend>

<?php

$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=3');

if ($row == "oletus") // Arvo 3
{
    echo '<p> <span class="virhe">Oletustekstiä ei
    ole muutettu hallintasivuilla.</span></p>';
}
else // Arvo 3
{
    // Luetaan funktiolla ylä valikon teksti viestit.php tiedostotsta
    $Yla_valikon_teksti = Yla_valikon_teksti();
    echo $Yla_valikon_teksti;
} // Arvo 3
?>
</legend>
<form method="post" action="index.php">
<?php
```



```
$Asiakaskysymys1 = Asiakaskysymys1();  
echo $Asiakaskysymys1;  
?>
```

```
<br>
```

```
<form method="post" action="index.php">
```

```
<select name="Valinta1">
```

```
<?php
```

```
$db = new SQLite3('tietokanta/valikko1.db');
```

```
$results = $db->query('SELECT VALIKKO, * from VALIKKO1');
```

```
while ($row = $results->fetchArray())
```

```
{
```

```
    echo "<option value='".$row['VALIKKO']."'>";
```

```
    echo ". $row['VALIKKO'] . '</option>'. "\n";
```

```
}
```

```
?>
```

```
</select>
```

```
<br>
```

```
<br>
```

```
<?php
```

```
$vastaus1 = $_POST['Valinta1'];
```

```
echo $vastaus1;
```

```
?>
```

```
<form method="post" action="index.php">
```

```
<?php
```

```
$Asiakaskysymys2 = Asiakaskysymys2();
```

```
echo $Asiakaskysymys2;
```

```
?>
```

```
<br>
```

```
<form method="post" action="index.php">
```

```
<select name="Valinta2">
```

```
<?php
```

```
$db = new SQLite3('tietokanta/valikko2.db');
```

```
$results = $db->query('SELECT VALIKKO, * from VALIKKO2');
```

```
while ($row = $results->fetchArray())
```

```
{
```

```
        echo "<option value='".$row['VALIKKO']."'>";
        echo ". $row['VALIKKO'] . '</option>'. "\n";
    }

?>
</select>
<br>
<br>

<?php
$Asiakaskysymys3 = Asiakaskysymys3();
echo $Asiakaskysymys3;
?>
<br>

<select name="Valinta3">
<?php
$db = new SQLite3('tietokanta/valikko3.db');
$results = $db->query('SELECT VALIKKO, * from VALIKKO3');
while ($row = $results->fetchArray())
    {
        echo "<option value='".$row['VALIKKO']."'>";
        echo ". $row['VALIKKO'] . '</option>'. "\n";
    }

?>
</select>
<br>
<br>

<?php
$Asiakaskysymys4 = Asiakaskysymys4();
echo $Asiakaskysymys4;
?>
<br>

<select name="Valinta4">
<?php
$db = new SQLite3('tietokanta/valikko4.db');
$results = $db->query('SELECT VALIKKO, * from VALIKKO4');
while ($row = $results->fetchArray())
```

```

        {
            echo "<option value='".$row['VALIKKO']."'>";
            echo ". $row['VALIKKO'] . '</option>'."."\\n";
        }
    ?>
</select>

<br>
<br>

<?php
$Asiakaskysymys5 = Asiakaskysymys5();
echo $Asiakaskysymys5;
?>
<br>

<select name="Valinta5">
<?php
$db = new SQLite3('tietokanta/valikko5.db');
$results = $db->query('SELECT VALIKKO, * from VALIKKO5');
while ($row = $results->fetchArray())
    {
        echo "<option value='".$row['VALIKKO']."'>";
        echo ". $row['VALIKKO'] . '</option>'."."\\n";
    }
?>
</select>

</fieldset>
<fieldset>

<legend>
<?php

$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT OLETUS, * FROM VIESTIT WHERE ID=4');

if ($row == "oletus") // Arvo 3
{
    echo '<p> <span class="virhe">Oletustekstiä ei

```

```
        ole muutettu hallintasivuilla.</span></p>';
    }
    else // Arvo 3
    {
        // Luetaan funktiolla ylä valikon teksti viestit.php tiedostotsta
        $Ala_valikon_teksti = Ala_valikon_teksti();
        echo $Ala_valikon_teksti;
    } // Arvo 3
?>
</legend>

<form method="post" action="index.php">
    <textarea name="text" cols="50" rows="10"></textarea>
    <p>
        <P class="center"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="tallenna"
class="submit"></P>
    </form>

</fieldset>
</div>
<!-- Asiakas kysymykset 1 HTML asetukset loppuu tähän-->
</div>

<!-- Asiakas kysymykset 1 PHP asetukset alkaa tästä-->
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $vastaus1 = $_POST['Valinta1'];
    $vastaus2 = $_POST['Valinta2'];
    $vastaus3 = $_POST['Valinta3'];
    $vastaus4 = $_POST['Valinta4'];
    $vastaus5 = $_POST['Valinta5'];
    $vastaus6 = $_POST['text'];

    if(isset($_POST['tallenna']))
    {
```

```
echo $_POST['Valinta1'];  
echo $vastaus2;  
echo $vastaus3;  
echo $vastaus4;  
echo $vastaus5;  
echo $vastaus1;
```

```
$db = new SQLite3('tietokanta/kysymykset.db');  
$row = $db->querySingle('SELECT KYSYMYYS1, * FROM KYSYMYKSET WHERE ID=1');  
$kysymys1 = $row;
```

```
$db = new SQLite3('tietokanta/kysymykset.db');  
$row = $db->querySingle('SELECT KYSYMYYS2, * FROM KYSYMYKSET WHERE ID=2');  
$kysymys2 = $row;
```

```
$db = new SQLite3('tietokanta/kysymykset.db');  
$row = $db->querySingle('SELECT KYSYMYYS3, * FROM KYSYMYKSET WHERE ID=3');  
$kysymys3 = $row;
```

```
$db = new SQLite3('tietokanta/kysymykset.db');  
$row = $db->querySingle('SELECT KYSYMYYS4, * FROM KYSYMYKSET WHERE ID=4');  
$kysymys4 = $row;
```

```
$db = new SQLite3('tietokanta/kysymykset.db');  
$row = $db->querySingle('SELECT KYSYMYYS5, * FROM KYSYMYKSET WHERE ID=5');  
$kysymys5 = $row;
```

```
$db = new SQLite3('tietokanta/PROFIILI_Oy.db');  
$db->exec("INSERT INTO ASIAKAS (VASTAUS1,VASTAUS2,VASTAUS3,VASTAUS4,VASTAUS5,VASTAUS6,  
KYSYMYYS1,KYSYMYYS2,KYSYMYYS3,KYSYMYYS4,KYSYMYYS5) VALUES  
('$vastaus1', '$vastaus2', '$vastaus3', '$vastaus4', '$vastaus5', '$vastaus6',  
'$kysymys1', '$kysymys2', '$kysymys3', '$kysymys4', '$kysymys5');");  
$db->close();
```

```
$db = new SQLite3('tietokanta/viestit.db');  
$db->exec("UPDATE VIESTIT SET OLETUS='on' WHERE ID=5");
```

```

$db->close();

$Selain_paivitys = selain_update();
echo $Selain_paivitys;
echo $arvo1;
}
if(isset($_POST['kirjaudu']))
{

$salasana=$_POST['salasana'];
$tunnus=$_POST['tunnus'];
$Salattu=md5($salasana);

$db = new SQLite3('tietokanta/talo.db');
$results = $db->query('SELECT * from KAYTTAJAT');
while ($row = $results->fetchArray())
{
    if ($row['KAYTTAJA'] == $tunnus)
    {

        if ($row['SALASANA'] == $Salattu)
        {

            $db->exec("UPDATE KAYTTAJAT SET TRACE='on' WHERE ID='2'");
            $db->exec("UPDATE KAYTTAJAT SET TRACE='ei' WHERE ID='1'");

        }

    }

}

$Selain_paivitys = selain_update();
echo $Selain_paivitys;

}

if(isset($_POST['ulos']))
{

$db = new SQLite3('tietokanta/talo.db');
$db->exec("UPDATE KAYTTAJAT SET TRACE='on' WHERE ID='1'");

```

```
$db->close();

$db = new SQLite3('tietokanta/talo.db');
$db->exec("UPDATE KAYTTAJAT SET TRACE='ei' WHERE ID='2'");
$db->close();

$Selain_paivitys = selain_update();
echo $Selain_paivitys;
}

}
?>

<!-- Asiakas kysymykset 1 PHP asetukset loppuu tähän-->

</form>
</body>
</html>
```

Liite 2 db\_luonti.php

```
<?php
$db = new SQLite3('tietokanta/PROFIILI_Oy.db');
$db->exec("CREATE TABLE IF NOT EXISTS ASIAKAS(
ID INTEGER PRIMARY KEY AUTOINCREMENT,
KYSYMYS1 CHAR(300),
KYSYMYS2 CHAR(300),
KYSYMYS3 CHAR(300),
KYSYMYS4 CHAR(300),
KYSYMYS5 CHAR(300),
VASTAUS1 CHAR(50),
VASTAUS2 CHAR(50),
VASTAUS3 CHAR(50),
VASTAUS4 CHAR(50),
VASTAUS5 CHAR(50),
VASTAUS6 CHAR(2000))");
?>

<?php
$db = new SQLite3('tietokanta/kysymykset.db');
$db->exec("CREATE TABLE IF NOT EXISTS KYSYMYKSET(
ID INT PRIMARY KEY NOT NULL,
KYSYMYS1 CHAR(300),
KYSYMYS2 CHAR(300),
KYSYMYS3 CHAR(300),
KYSYMYS4 CHAR(300),
KYSYMYS5 CHAR(300),
KYSYMYS6 CHAR(300))");
$db->exec("INSERT INTO KYSYMYKSET (ID,KYSYMYS1)
VALUES ('1','Asiakkaan kysymys 1 puuttuu')");
$db->exec("INSERT INTO KYSYMYKSET (ID,KYSYMYS2)
VALUES ('2','Asiakkaan kysymys 2 puuttuu')");
$db->exec("INSERT INTO KYSYMYKSET (ID,KYSYMYS3)
VALUES ('3','Asiakkaan kysymys 3 puuttuu')");
$db->exec("INSERT INTO KYSYMYKSET (ID,KYSYMYS4)
VALUES ('4','Asiakkaan kysymys 4 puuttuu')");
$db->exec("INSERT INTO KYSYMYKSET (ID,KYSYMYS5)
VALUES ('5','Asiakkaan kysymys 5 puuttuu')");
$db->close();
```



?>

<?php

```
$db = new SQLite3('tietokanta/talo.db');
$db->exec("CREATE TABLE IF NOT EXISTS KAYTTAJAT(
ID INT PRIMARY KEY NOT NULL,
KAYTTAJA CHAR(300),
SALASANA CHAR(600),
TRACE CHAR(300))");
$salasana = "salalomake";
$Salattu_salasana = md5($salasana);
$db->exec("INSERT OR IGNORE INTO KAYTTAJAT (ID,KAYTTAJA,SALASANA,TRACE)
VALUES ('1','asiakas','on')");
$db->exec("INSERT OR IGNORE INTO KAYTTAJAT (ID,KAYTTAJA,SALASANA,TRACE)
VALUES ('2','hallinta','$Salattu_salasana','ei')");
$db->close();
```

?>

<?php

```
$db = new SQLite3('tietokanta/viestit.db');
$db->exec("CREATE TABLE IF NOT EXISTS VIESTIT (
ID INT PRIMARY KEY NOT NULL,
VIESTI CHAR(500),
OLETUS CHAR(10))");
$db->close();
```

?>

<?php

```
$db = new SQLite3('tietokanta/viestit.db');
$db->exec("INSERT OR IGNORE INTO VIESTIT (ID,VIESTI,OLETUS) VALUES
('1','Alkutervehdys asiakkaalle tähän','oletus')");
$db->exec("INSERT OR IGNORE INTO VIESTIT (ID,VIESTI,OLETUS) VALUES
('2','Palute viesti asiakkaalle tähän','oletus')");
$db->exec("INSERT OR IGNORE INTO VIESTIT (ID,VIESTI,OLETUS) VALUES
('3','Ylävalikon selitys asiakkaalle tähän','oletus')");
$db->exec("INSERT OR IGNORE INTO VIESTIT (ID,VIESTI,OLETUS) VALUES
('4','Alavalikon selitys asiakkaalle tähän','oletus')");
$db->exec("INSERT OR IGNORE INTO VIESTIT (ID,VIESTI,OLETUS) VALUES
('5','tyhjä','ei')");
$db->close();
```

?>

<?php

```
$db = new SQLite3('tietokanta/valikko1.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO1 (
ID INT PRIMARY KEY NOT NULL,
VALIKKO CHAR(20))");
$db->close();
?>

<?php
$db = new SQLite3('tietokanta/valikko1_lisaa.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO1_LISAA (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
VALIKKO CHAR(20))");
$db->exec("INSERT OR IGNORE INTO VALIKKO1_LISAA (ID,VALIKKO) VALUES
('1','Valitse')");
$db->close();
?>

<?php
$db = new SQLite3('tietokanta/valikko2.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO2 (
ID INT PRIMARY KEY NOT NULL,
VALIKKO CHAR(20))");
$db->close();
?>

<?php
$db = new SQLite3('tietokanta/valikko2_lisaa.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO2_LISAA (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
VALIKKO CHAR(20))");
$db->exec("INSERT OR IGNORE INTO VALIKKO2_LISAA (ID,VALIKKO) VALUES
('1','Valitse')");
$db->close();
?>

<?php
$db = new SQLite3('tietokanta/valikko3.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO3 (
ID INT PRIMARY KEY NOT NULL,
VALIKKO CHAR(20))");
$db->close();
?>

<?php
```

```
$db = new SQLite3('tietokanta/valikko3_lisaa.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO3_LISAA (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
VALIKKO CHAR(20))");
$db->exec("INSERT OR IGNORE INTO VALIKKO3_LISAA (ID,VALIKKO) VALUES
('1','Valitse')");
$db->close();
?>
```

```
<?php
```

```
$db = new SQLite3('tietokanta/valikko4.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO4 (
ID INT PRIMARY KEY NOT NULL,
VALIKKO CHAR(20))");
$db->close();
?>
```

```
<?php
```

```
$db = new SQLite3('tietokanta/valikko4_lisaa.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO4_LISAA (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
VALIKKO CHAR(20))");
$db->exec("INSERT OR IGNORE INTO VALIKKO4_LISAA (ID,VALIKKO) VALUES
('1','Valitse')");
$db->close();
?>
```

```
<?php
```

```
$db = new SQLite3('tietokanta/valikko5.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO5 (
ID INT PRIMARY KEY NOT NULL,
VALIKKO CHAR(20))");
$db->close();
?>
```

```
<?php
```

```
$db = new SQLite3('tietokanta/valikko5_lisaa.db');
$db->exec("CREATE TABLE IF NOT EXISTS VALIKKO5_LISAA (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
VALIKKO CHAR(20))");
$db->exec("INSERT OR IGNORE INTO VALIKKO5_LISAA (ID,VALIKKO) VALUES
('1','Valitse')");
$db->close();
```

?>

## Liite 3 Asiakaskysymykset\_luku.php

```
<?php
function Asiakaskysymys1 ()
{
$db = new SQLite3('tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYS1, * FROM KYSYMYKSET WHERE ID=1');
echo $row;
}
function Asiakaskysymys2 ()
{
$db = new SQLite3('tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYS2, * FROM KYSYMYKSET WHERE ID=2');
echo $row;
}
function Asiakaskysymys3 ()
{
$db = new SQLite3('tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYS3, * FROM KYSYMYKSET WHERE ID=3');
echo $row;
}
function Asiakaskysymys4 ()
{
$db = new SQLite3('tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYS4, * FROM KYSYMYKSET WHERE ID=4');
echo $row;
}
function Asiakaskysymys5 ()
{
$db = new SQLite3('tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYS5, * FROM KYSYMYKSET WHERE ID=5');
echo $row;
}
?>
```

Liite 4      selain\_update.php

```
<?php
function selain_update ()
{
    echo '
<html>
<head>
<meta http-equiv=Refresh content="3; url='.$_SERVER['PHP_SELF'].'">
</head>
</html>';
}
?>
```

## Liite 5 viestit.php

```
<?php
function Asiakkaan_alkutervehdys ()
{
$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=1');
echo $row;
$db->close();
}
function Palaute_asiakkaalle ()
{
$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=2');
echo $row;
}
function Yla_valikon_teksti ()
{
$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=3');
echo $row;
}
function Ala_valikon_teksti ()
{
$db = new SQLite3('tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=4');
echo $row;
}
function VIESTI_TOTTA_ei ()
{
$db = new SQLite3('tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=5");
$db->close();
}
?>
```

## Liite 6      hallinta/Excel\_function.php

```
<?php
/* Luetaan excel tiedostot funktioiden avulla*/
function Excel1 ()
{
    session_start();
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS1, COUNT(*) from ASIAKAS');
    while ($row = $results->fetchArray())
    {
        $_SESSION["numero01"] = $row['COUNT(*)'];
    }

    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS1, ID, KYSYMYYS1, COUNT(*) from ASIAKAS GROUP BY
    VASTAUS1');
    while ($row = $results->fetchArray())
    {
        header( "Content-Type: application/vnd.ms-excel" );
        header( "Content-disposition: attachment; filename=raportti1.xls" );
        if (empty($counter))
        {
            $counter = 1;
            echo "ID \t KYSYMYYS2 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
        }
        else{}
        echo $row['ID'] . "\t" . $row['KYSYMYYS1'] . "\t" . $row['VASTAUS1'] . "\t" . $row['COUNT(*)'] . "\t"
        . "\t" . round($row['COUNT(*)'] / $_SESSION["numero01"] * 100, 0) . "\n";
    }
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS1, COUNT(*) from ASIAKAS');
    while ($row = $results->fetchArray())
    {
        echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
    }
    session_unset();
    session_destroy();
}
```



```
function Excel2 ()
{
    session_start();
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS2, COUNT(*) from ASIAKAS');
    while ($row = $results->fetchArray())
    {
        $_SESSION["numero02"] = $row['COUNT(*)'];
    }

    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS2, ID, KYSYMYYS2, COUNT(*) from ASIAKAS GROUP BY VASTAUS2');
    while ($row = $results->fetchArray())
    {
        header( "Content-Type: application/vnd.ms-excel" );
        header( "Content-disposition: attachment; filename=raportti2.xls" );
        if (empty($counter))
        {
            $counter = 1;
            echo "ID \t KYSYMYYS2 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
        }
        else{}
        echo $row['ID'] . "\t" . $row['KYSYMYYS2'] . "\t" . $row['VASTAUS2'] . "\t" . $row['COUNT(*)'] . "\t" . "\t" . round($row['COUNT(*)'] / $_SESSION["numero02"] * 100, 0) . "%\n";
    }
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS2, COUNT(*) from ASIAKAS');
    while ($row = $results->fetchArray())
    {
        echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
    }
    session_unset();
    session_destroy();
}

function Excel3 ()
{
```

```

session_start();
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS3, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    $_SESSION["numero03"] = $row['COUNT(*)'];
}

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS3, ID, KYSYMYSS3, COUNT(*) from ASIAKAS GROUP BY
VASTAUS3');
while ($row = $results->fetchArray())
{
    header( "Content-Type: application/vnd.ms-excel" );
    header( "Content-disposition: attachment; filename=raportti3.xls" );
    if (empty($counter))
    {
        $counter = 1;
        echo "ID \t KYSYMYSS3 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    else{}
    echo $row['ID'] . "\t" . $row['KYSYMYSS3'] . "\t" . $row['VASTAUS3'] . "\t" . $row['COUNT(*)'] . "\t"
    . "\t" . round($row['COUNT(*)'] / $_SESSION["numero03"] * 100, 0) . "%\n";
}
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS3, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)'] . "\n";
}
session_unset();
session_destroy();
}

function Excel4 ()
{
    session_start();
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS4, COUNT(*) from ASIAKAS');

```

```
while ($row = $results->fetchArray())
{
    $_SESSION["numero04"] = $row['COUNT(*)'];
}

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS4, ID, KYSYMYYS4, COUNT(*) from ASIAKAS GROUP BY VASTAUS4');
while ($row = $results->fetchArray())
{
    header( "Content-Type: application/vnd.ms-excel" );
    header( "Content-disposition: attachment; filename=raportti4.xls" );
    if (empty($counter))
    {
        $counter = 1;
        echo "ID \t KYSYMYYS4 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    else{}
    echo $row['ID'] . "\t" . $row['KYSYMYYS4'] . "\t" . $row['VASTAUS4'] . "\t" . $row['COUNT(*)'] . "\t" . "\t" . round($row['COUNT(*)'] / $_SESSION["numero04"] * 100, 0) . "%\n";
}

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS4, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
}
session_unset();
session_destroy();
}

function Excel5 ()
{
    session_start();
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT VASTAUS5, COUNT(*) from ASIAKAS');
    while ($row = $results->fetchArray())
    {
```

```

$_SESSION["numero05"] = $row['COUNT(*)'];
}

```

```

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS5, ID, KYSYMY5, COUNT(*) from ASIAKAS GROUP BY
VASTAUS5');
while ($row = $results->fetchArray())
{
    header( "Content-Type: application/vnd.ms-excel" );
    header( "Content-disposition: attachment; filename=raportti5.xls" );
    if (empty($counter))
    {
        $counter = 1;
        echo "ID \t KYSYMY5 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    else{}
    echo $row['ID'] . "\t" . $row['KYSYMY5'] . "\t" . $row['VASTAUS5'] . "\t" . $row['COUNT(*)'] . "\t"
    . "\t" . round($row['COUNT(*)'] / $_SESSION["numero05"] * 100, 0) . "\n";
}
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS5, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)'] . "\n";
}
session_unset();
session_destroy();
}

```

```

function Excel6 ()
{
    $db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
    $results = $db->query('SELECT * from ASIAKAS');
    while ($row = $results->fetchArray())
    {
        header( "Content-Type: application/vnd.ms-excel" );
        header( "Content-disposition: attachment; filename=raportti6.xls" );
        if (empty($counter))

```

```
{
$counter = 1;
echo "ID \t VAPAA VASTAAMINEN\n ";
}
else{}
echo $row['ID'] . "\t" . $row['VASTAUS6'] . "\n";
}
}
```

```
function Excel7 ()
```

```
{
session_start();
header("Content-type: application/vnd.ms-excel;");
header("Content-Disposition: attachment;Filename=raportti7.xls");
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS1, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    $_SESSION["numero1"] = $row['COUNT(*)'];
}
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS1, ID, KYSYMYS1, COUNT(*) from ASIAKAS GROUP BY VASTAUS1');
while ($row = $results->fetchArray())
{
    if (empty($counter))
    {
        $counter = 1;
        echo "ID \t KYSYMYS1 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    echo $row['ID'] . "\t" . $row['KYSYMYS1'] . "\t" . $row['VASTAUS1'] . "\t" . $row['COUNT(*)'] . "\t" . "\t" . round($row['COUNT(*)'] / $_SESSION["numero1"] * 100, 0) . "\n";
}
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS1, COUNT(*) from ASIAKAS');
```

```
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
}
echo "\n";
echo "\n";
$db->close();
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS2, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    $_SESSION["numero2"] = $row['COUNT(*)'];
}
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS2, ID, KYSYMYKSET, COUNT(*) from ASIAKAS GROUP BY VASTAUS2');
while ($row = $results->fetchArray())
{
    if ($counter == 1)
    {
        $counter = 2;
        echo "ID \t KYSYMYKSET \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    echo $row['ID'] . "\t" . $row['KYSYMYKSET'] . "\t" . $row['VASTAUS2'] . "\t" . $row['COUNT(*)'] . "\t" . "\t" . round($row['COUNT(*)'] / $_SESSION["numero2"] * 100, 0) . "\n";
}
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS2, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
}
```

```
echo "\n";
```

```
echo "\n";
$db->close();
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS3, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    $_SESSION["numero3"] = $row['COUNT(*)'];
}
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS3, ID, KYSYMYSS3, COUNT(*) from ASIAKAS GROUP BY
VASTAUS3');
while ($row = $results->fetchArray())
{
    if ($counter == 2)
    {
        $counter = 3;
        echo "ID \t KYSYMYSS3 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    echo $row['ID'] . "\t" . $row['KYSYMYSS3'] . "\t" . $row['VASTAUS3'] . "\t" . $row['COUNT(*)'] . "\t"
    . "\t" . round($row['COUNT(*)'] / $_SESSION["numero3"] * 100, 0) . "\n";
}
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS3, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
}
echo "\n";
echo "\n";
$db->close();
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS4, COUNT(*) from ASIAKAS');
```

```
while ($row = $results->fetchArray())
{
    $_SESSION["numero4"] = $row['COUNT(*)'];
}

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS4, ID, KYSYMYSS4, COUNT(*) from ASIAKAS GROUP BY VASTAUS4');
while ($row = $results->fetchArray())
{
    if ($counter == 3)
    {
        $counter = 4;
        echo "ID \t KYSYMYSS4 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    echo $row['ID'] . "\t" . $row['KYSYMYSS4'] . "\t" . $row['VASTAUS4'] . "\t" . $row['COUNT(*)'] . "\t" . "\t" . round($row['COUNT(*)'] / $_SESSION["numero4"] * 100, 0) . "%\n";
}

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS4, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)'] . "\n";
}
echo "\n";
echo "\n";
$db->close();

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS5, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    $_SESSION["numero5"] = $row['COUNT(*)'];
}

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
```



```
$results = $db->query('SELECT VASTAUS5, ID, KYSYMYYS5, COUNT(*) from ASIAKAS GROUP BY
VASTAUS5');
while ($row = $results->fetchArray())
{
    if ($counter == 4)
    {
        $counter = 5;
        echo "ID \t KYSYMYYS5 \t VASTAUKSET \t SAMOJA \t VASTAUKSIA \t PROSENTTIA %\n ";
    }
    echo $row['ID'] . "\t" . $row['KYSYMYYS5'] . "\t" . $row['VASTAUS5'] . "\t" . $row['COUNT(*)'] . "\t"
    . "\t" . round($row['COUNT(*)'] / $_SESSION["numero5"] * 100, 0) . "\n";
}
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT VASTAUS5, COUNT(*) from ASIAKAS');
while ($row = $results->fetchArray())
{
    echo "\t" . "\t" . "\t" . "\t" . $row['COUNT(*)']. "\n";
}

echo "\n";
echo "\n";
$db->close();

$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query("SELECT * from ASIAKAS");
while ($row = $results->fetchArray())
{
    if ($counter == 5)
    {
        $counter = 6;
        echo "ID \t VAPAA VASTAAMINEN\n ";
    }
    echo $row['ID'] . "\t" . $row['VASTAUS6'] . "\n";
}
echo "\n";
echo "\n";

$db->close();
session_unset();
```

```
session_destroy();  
}  
?>
```

## Liite 7      hallinta/Excel\_valinnat.php

```
<?php
require "Excel_function.php";
?>
```

```
<?php
if(isset($_POST['tulosta'])){
    $VALIKKO = $_POST['VALIKKO'];
    if ($VALIKKO == "VASTAUS1")
    {
        $Excel1 = Excel1();
    }
    if ($VALIKKO == "VASTAUS2")
    {
        $Excel2 = Excel2();
    }
    if ($VALIKKO == "VASTAUS3")
    {
        $Excel3 = Excel3();
    }
    if ($VALIKKO == "VASTAUS4")
    {
        $Excel4 = Excel4();
    }
    if ($VALIKKO == "VASTAUS5")
    {
        $Excel5 = Excel5();
    }
    if ($VALIKKO == "VASTAUS6")
    {
        $Excel6 = Excel6();
    }
    if ($VALIKKO == "VASTAUS7")
    {
        $Excel7 = Excel7();
    }
}
?>
```

## Liite 8      hallinta/hallinta\_kysymykset\_luku.php

```
<?php
function Asiakaskysymys1 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=1');
echo $row;
}
function Asiakaskysymys2 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=2');
echo $row;
}
function Asiakaskysymys3 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=3');
echo $row;
}
function Asiakaskysymys4 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=4');
echo $row;
}
function Asiakaskysymys5 ()
{
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=5');
echo $row;
}
?>
```

## Liite 9      hallinta/poisto.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO 8859-1;" />
<link rel='stylesheet' type='text/css' media='all' href='css/sivut.css' />
<style type='text/css' media='screen'>@import "css/sivut.css";</style>
<title>Tietokanta</title>
</style>
</head>
<body>

<div id="wrapper">
<div class="yla_kuva">

</div>
<!-- <div class="yla_kuva"> LOPPUU TÄHÄN-->
<div class="linkki_laatikko">
<div class="poisto">
<ul id="linkit">
<li id="taulukot"><a href="hallinta.php" title="Taulukot">TAULUKOT</a></li>
<li id="ruksit"><a href="ruksit.php" title="Ruksit">RUKSIT</a></li>
<li id="poisto"><a href="poisto.php" title="Poisto">TIETOKANTA</a></li>
</ul>
</div>
<!-- <div class="poisto"> LOPPUU TÄHÄN-->
</div>
<!-- <div class="linkki_laatikko"> LOPPUU TÄHÄN-->
<div class="ala_sisalto">
<!-- <div class="ala_sisalto"> Taulukoiden tulokset -->
<fieldset>
<fieldset id="sivusto">
<BR>
<BR>
<h2>Poista kaikki tietokannat</h2>
<BR>
```

```
<?php
require "Taulukot_poisto.php";
$Taulukot1_poisto = Taulukot1_poisto();
echo $Taulukot1_poisto;
echo '<br>';
echo '<br>';
?>
<br>
<br>
<br>
</div>
<!-- <div class="ala_sisalto"> LOPPUU TÄHÄN-->
</div>
<!-- <div id="wrapper"> LOPPUU TÄHÄN-->
</form>
</body>
</html>
```

## Liite 10      hallinta/taulukot.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel='stylesheet' type='text/css' media='all' href='css/sivut.css' />
<style type='text/css' media='screen'>@import "css/sivut.css";</style>
<title>TAULUKOT</title>
</style>
</head>
<body>
<div id="wrapper">
<div id="kirjautuminen">
<div id="oikea">
<?php
$db = new SQLite3('../tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
if ($row['TRACE'] == "on")
{
echo '<label for="salasana">Kirjautunut: </label>';
echo "<td>" . $row['KAYTTAJA'] . "</td>";
}
}
?>
</div>
</div>
<div id="Logo">

</div>
<div class="linkki_laatikko">
<div class="taulukot">
<ul id="linkit">
<li id="palaute"><a href="../index.php" title="Palaute">PALAUTE</a></li>
```

```
<li id="taulukot"><a href="taulukot.php" title="Taulukot">TAULUKOT</a></li>
<li id="valikot"><a href="valikot.php" title="Ruksit">VALIKOT</a></li>
<li id="viestit"><a href="viestit.php" title="Viestit">VIESTIT</a></li>
<li id="tietokanta"><a href="tietokanta.php" title="Tietokanta">TIETOKANTA</a></li>
</ul>
</div>
</div>
<?php
require "hallinta_kysymykset_luku.php";
?>
<!-- Asiakas kysymykset 1 HTML asetukset alkaa tästä-->
<fieldset>
<div id="as_1-5">
<fieldset>
<legend>Asiakas kysymys 1 - 5 vastaukset</legend>
<?php
require "../selain_update.php";
require "taulukot_funktiot.php";
$taulukot_uusi = taulukot_uusi();
echo $taulukot_uusi;
?>
</fieldset>
</div>
<div id="Asiakas_kysymys">
<fieldset>
<legend>Asiakas kysymys 6 vastaukset</legend>
<?php
$taulukot_uusi2 = taulukot_uusi2();
echo $taulukot_uusi2;
?> </textarea>
</fieldset>
</div>
<div id="vastauksen_lisaaminen">
<fieldset>
<legend>Tulostus valikko</legend>
<form method="post" action="Excel_valinnat.php">
<label id="VALIKKO" for="VALIKKO">Tulostuksen vaihtoehto:</label>
<select name="VALIKKO" id="VALIKKO">
<option value="1">Valikoi vaihtoehto</option>
```



```
<option value="VASTAUS1">Tulosta Excel tiedostoon vain vastaus 1</option>
<option value="VASTAUS2">Tulosta Excel tiedostoon vain vastaus 2</option>
<option value="VASTAUS3">Tulosta Excel tiedostoon vain vastaus 3</option>
<option value="VASTAUS4">Tulosta Excel tiedostoon vain vastaus 4</option>
<option value="VASTAUS5">Tulosta Excel tiedostoon vain vastaus 5</option>
<option value="VASTAUS6">Tulosta Excel tiedostoon vain vastaus 6</option>
<option value="VASTAUS7">Tulosta Excel tiedostoon kaikki vastaukset</option>
</select>

<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="tulosta" NAME="tulosta" class="submit"></P>
</p>
</form>
</fieldset>
</div>
<!-- Asiakas kysymykset 1 HTML asetukset loppuu tähän-->
</div>
<!-- Asiakas kysymykset 1 PHP asetukset alkaa tästä-->
</form>
</body>
</html>
```

## Liite 11      hallinta/taulukot\_funktiot.php

```
<?php
/* function Taulukot () luetaan asiakas ja hallinta sivustolla */
function Valikko1 ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";
$db = new SQLite3('../tietokanta/valikko1.db');
$results = $db->query('SELECT * from VALIKKO1');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko1_id' value='" . $row['ID'] . "'>
<input type='hidden' name='val1' value='" . $row['VALIKKO'] . "'>
<input type='submit' name='valikko1_poista' value='poista'>
</form>
</td>";
echo "</tr>";
}
echo "</table> ";
}
/* function Taulukot () luetaan asiakas ja hallinta sivustolla */
function Valikko1_lisaa ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
```

```

<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";
$db = new SQLite3('../tietokanta/valikko1_lisaa.db');
$results = $db->query('SELECT * from VALIKKO1_LISAA');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko1_lisaa_id' value='" . $row['ID'] . "'>
<input type='hidden' name='valikko1_lisaa_kysymys' value='" . $row['VALIKKO'] . "'>
<input type='submit' name='valikko1_lisaa_poista' value='poista'>
<input type='submit' name='valikko1_lisaa' value='Lisää'>
</form>
</td>";
echo "</tr>";
}
echo '</table> ';
}

```

```

function Valikko2 ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";
$db = new SQLite3('../tietokanta/valikko2.db');
$results = $db->query('SELECT * from VALIKKO2');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko2_id' value='" . $row['ID'] . "'>

```

```
<input type='hidden' name='val2' value=''' . $row['VALIKKO'] . ''>
<input type='submit' name='valikko2_poista' value='poista'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
function Valikko2_lisaa ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";

$db = new SQLite3('../tietokanta/valikko2_lisaa.db');
$results = $db->query('SELECT * from VALIKKO2_LISAA');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko2_lisaa_id' value=''' . $row['ID'] . ''>
<input type='hidden' name='valikko2_lisaa_kysymys' value=''' . $row['VALIKKO'] . ''>
<input type='submit' name='valikko2_lisaa_poista' value='poista'>
<input type='submit' name='valikko2_lisaa' value='Lisää'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
function Valikko3 ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
```

```

<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";

$db = new SQLite3('../tietokanta/valikko3.db');
$results = $db->query('SELECT * from VALIKKO3');
while ($row = $results->fetchArray())
{
    echo "<tr>";
    echo "<td>" . $row['ID'] . "</td>";
    echo "<td>" . $row['VALIKKO'] . "</td>";
    echo "<td><form name='css2' action='valikot.php' method='post'>
    <input type='hidden' name='valikko3_id' value='" . $row['ID'] . "'>
    <input type='hidden' name='val3' value='" . $row['VALIKKO'] . "'>
    <input type='submit' name='valikko3_poista' value='poista'>
    </form>
    </td>";
    echo "</tr>";
}
echo '</table> ';
}

function Valikko3_lisaa ()
{
    echo "<table bgcolor='grey' width=300 border='1'>
    <tr>
    <th><align=left><font color=white>ID</font></th>
    <th><align=left><font color=white>VALIKKO</font></th>
    <th><align=left><font color=white>TOIMINNOT</font></th>
    </tr>";

    $db = new SQLite3('../tietokanta/valikko3_lisaa.db');
    $results = $db->query('SELECT * from VALIKKO3_LISAA');
    while ($row = $results->fetchArray())
    {
        echo "<tr>";
        echo "<td>" . $row['ID'] . "</td>";
        echo "<td>" . $row['VALIKKO'] . "</td>";
        echo "<td><form name='css2' action='valikot.php' method='post'>

```

```

<input type='hidden' name='valikko3_lisaa_id' value='' . $row['ID'] . "'>
<input type='hidden' name='valikko3_lisaa_kysymys' value='' . $row['VALIKKO'] . "'>
<input type='submit' name='valikko3_lisaa_poista' value='poista'>
<input type='submit' name='valikko3_lisaa' value='Lisää'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
function Valikko4 ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";

$db = new SQLite3('../tietokanta/valikko4.db');
$results = $db->query('SELECT * from VALIKKO4');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko4_id' value='' . $row['ID'] . "'>
<input type='hidden' name='val4' value='' . $row['VALIKKO'] . "'>
<input type='submit' name='valikko4_poista' value='poista'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
function Valikko4_lisaa ()
{
echo "<table bgcolor='grey' width=300 border='1'>

```

```
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";
$db = new SQLite3('../tietokanta/valikko4_lisaa.db');
$results = $db->query('SELECT * from VALIKKO4_LISAA');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko4_lisaa_id' value='" . $row['ID'] . "'>
<input type='hidden' name='valikko4_lisaa_kysymys' value='" . $row['VALIKKO'] . "'>
<input type='submit' name='valikko4_lisaa_poista' value='poista'>
<input type='submit' name='valikko4_lisaa' value='Lisää'>
</form>
</td>";
echo "</tr>";
}
echo "</table> ";
}
function Valikko5 ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";

$db = new SQLite3('../tietokanta/valikko5.db');
$results = $db->query('SELECT * from VALIKKO5');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
```

```

echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko5_id' value='' . $row['ID'] . ''>
<input type='hidden' name='val5' value='' . $row['VALIKKO'] . ''>
<input type='submit' name='valikko5_poista' value='poista'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
function Valikko5_lisaa ()
{
echo "<table bgcolor='grey' width=300 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VALIKKO</font></th>
<th><align=left><font color=white>TOIMINNOT</font></th>
</tr>";
$db = new SQLite3('../tietokanta/valikko5_lisaa.db');
$results = $db->query('SELECT * from VALIKKO5_LISAA');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VALIKKO'] . "</td>";
echo "<td><form name='css2' action='valikot.php' method='post'>
<input type='hidden' name='valikko5_lisaa_id' value='' . $row['ID'] . ''>
<input type='hidden' name='valikko5_lisaa_kysymys' value='' . $row['VALIKKO'] . ''>
<input type='submit' name='valikko5_lisaa_poista' value='poista'>
<input type='submit' name='valikko5_lisaa' value='Lisää'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
?>

<?php

```



```
/* function Taulukot () luetaan asiakas ja hallinta sivustolla */
function taulukot_uusi ()
{
echo '<div style="width: 700px; overflow: auto;">';
echo "<table bgcolor='purple' border='1'>
<tr>
<th><align=left><font color=white>KYSYMYS1</font></th>
<th><align=left><font color=white>VASTAUS1</font></th>
<th><align=left><font color=purple>xx</font></th>
<th><align=left><font color=white>KYSYMYS2</font></th>
<th><align=left><font color=white>VASTAUS2</font></th>
<th><align=left><font color=purple>xx</font></th>
<th><align=left><font color=white>KYSYMYS3</font></th>
<th><align=left><font color=white>VASTAUS3</font></th>
<th><align=left><font color=purple>xx</font></th>
<th><align=left><font color=white>KYSYMYS4</font></th>
<th><align=left><font color=white>VASTAUS4</font></th>
<th><align=left><font color=purple>xx</font></th>
<th><align=left><font color=white>KYSYMYS5</font></th>
<th><align=left><font color=white>VASTAUS5</font></th>
</tr>";
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT * from ASIAKAS');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['KYSYMYS1'] . "</td>";
echo "<td>" . $row['VASTAUS1'] . "</td>";
echo "<td>" . " . "</td>";
echo "<td>" . $row['KYSYMYS2'] . "</td>";
echo "<td>" . $row['VASTAUS2'] . "</td>";
echo "<td>" . " . "</td>";
echo "<td>" . $row['KYSYMYS3'] . "</td>";
echo "<td>" . $row['VASTAUS3'] . "</td>";
echo "<td>" . " . "</td>";
echo "<td>" . $row['KYSYMYS4'] . "</td>";
echo "<td>" . $row['VASTAUS4'] . "</td>";
echo "<td>" . " . "</td>";
echo "<td>" . $row['KYSYMYS5'] . "</td>";
```

```
echo "<td>" . $row['VASTAUS5'] . "</td>";
echo "<td><form name='css2' action='taulukot.php' method='post'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
?>
```

```
<?php
/* function Taulukot () luetaan asiakas ja hallinta sivustolla */
function taulukot_uusi2 ()
{
echo "<table bgcolor='purple' width=200 border='1'>
<tr>
<th><align=left><font color=white>ID</font></th>
<th><align=left><font color=white>VASTAUS6</font></th>
</tr>";
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$results = $db->query('SELECT * from ASIAKAS');
while ($row = $results->fetchArray())
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['VASTAUS6'] . "</td>";
echo "<td><form name='css2' action='taulukot.php' method='post'>
</form>
</td>";
echo "</tr>";
}
echo' </table> ';
}
?>
```

## Liite 12      hallinta/taulukot\_poisto.php

```
<form id="Taulukot" action="" method="post">
</p>
<P class="center"><INPUT TYPE="SUBMIT" VALUE="Poista taulukot" NAME="kanta_poisto"
class="submit"></P>
```

```
<?php
function Taulukot1_poisto ()
{
if(isset($_POST['kanta_poisto']))
{
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$db->exec("DROP TABLE IF EXISTS ASIAKAS");
if(!$db)
{
echo $db->lastErrorMsg();
}
$db->close();
```

```
$db = new SQLite3('../tietokanta/PROFIILI_Oy.db');
$db->exec("DROP TABLE IF EXISTS HALLINTA");
if(!$db)
{
echo $db->lastErrorMsg();
}
$db->close();
```

```
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("DROP TABLE IF EXISTS VIESTIT");
if(!$db)
{
echo $db->lastErrorMsg();
}
$db->close();
```

```
$db = new SQLite3('../tietokanta/kysymykset.db');  
$db->exec("DROP TABLE IF EXISTS KYSYMYKSET");  
if(!$db)  
{  
    echo $db->lastErrorMsg();  
}  
$db->close();
```

```
$db = new SQLite3('../tietokanta/talo.db');  
$db->exec("DROP TABLE IF EXISTS KAYTTAJAT");  
if(!$db)  
{  
    echo $db->lastErrorMsg();  
}  
$db->close();
```

```
$db = new SQLite3('../tietokanta/valikko1.db');  
$db->exec("DROP TABLE IF EXISTS VALIKKO1");  
if(!$db)  
{  
    echo $db->lastErrorMsg();  
}  
$db->close();
```

```
$db = new SQLite3('../tietokanta/valikko2.db');  
$db->exec("DROP TABLE IF EXISTS VALIKKO2");  
if(!$db)  
{  
    echo $db->lastErrorMsg();  
}  
$db->close();
```

```
$db = new SQLite3('../tietokanta/valikko3.db');  
$db->exec("DROP TABLE IF EXISTS VALIKKO3");  
if(!$db)  
{  
    echo $db->lastErrorMsg();  
}  
$db->close();
```

```
$db = new SQLite3('../tietokanta/valikko4.db');
$db->exec("DROP TABLE IF EXISTS VALIKKO4");
if(!$db)
{
    echo $db->lastErrorMsg();
}
$db->close();

$db = new SQLite3('../tietokanta/valikko5.db');
$db->exec("DROP TABLE IF EXISTS VALIKKO5");
if(!$db)
{
    echo $db->lastErrorMsg();
}
$db->close();
echo '<br>';
echo '<br>';
echo ' Kaikki tietokannat poistettu onnistuneesti.';
echo '
<html>
<head>
<meta http-equiv=Refresh content="3; url='.$_SERVER['PHP_SELF'].'">
</head>
</html>';
}
}
?>
</form>
```

## Liite 13      hallinta/tietokanta.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel='stylesheet' type='text/css' media='all' href='css/sivut.css' />
<style type='text/css' media='screen'>@import "css/sivut.css";</style>
<title>TIETOKANTA</title>

</style>
</head>

<body>
<div id="wrapper">
<div id="kirjautuminen">
<div id="oikea">
<?php
$db = new SQLite3('../tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
if ($row["TRACE"] == "on")
{
echo '<label for="salasana">Kirjautunut: </label>';
echo "<td>" . $row['KAYTTAJA'] . "</td>";
}
}
?>
</div>
</div>

<div id="Logo">

</div>
<div class="linkki_laatikko">
```

```
<div class="tietokanta">
<ul id="linkit">
<li id="palaute"><a href="../index.php" title="Palaute">PALAUTE</a></li>
<li id="taulukot"><a href="taulukot.php" title="Taulukot">TAULUKOT</a></li>
<li id="valikot"><a href="valikot.php" title="Ruksit">VALIKOT</a></li>
<li id="viestit"><a href="viestit.php" title="Viestit">VIESTIT</a></li>
<li id="tietokanta"><a href="tietokanta.php" title="Tietokanta">TIETOKANTA</a></li>
</ul>
</div>
```

```
</div>
<!-- <div class="linkki_laatikko"> LOPPUU TÄHÄN-->
<!-- Asiakas kysymykset 1 HTML asetukset alkaa tästä-->
<fieldset>
<div id="Asiakas_kysymys">
<fieldset>
<legend>Poista kaikki tietokannat</legend>
<?php
require "Taulukot_poisto.php";
$Taulukot1_poisto = Taulukot1_poisto();
echo $Taulukot1_poisto;
?>
</fieldset>
</div>
<div id="vastauksen_lisaaminen">
<form method="post" action="tietokanta.php">
<fieldset>
<legend>Vaihda salasana</legend>
<p>
<label for="salasana">Salasana:</label>
<input type="password" name="salasana" value=""
</p>
<p>
<label for="salasana2">Uudestaan:</label>
<input type="password" name="salasana2" value="" />
</p>
<p>
<P><INPUT TYPE="SUBMIT" VALUE="Vaihda" NAME="vaihda"></P>
```

```
</p>
<?php
$eka = $_POST['salasana'];
$toka = $_POST['salasana2'];
if(isset($_POST['vaihda']))
{
if (empty($eka) || empty($toka))
{
echo'Kenttä ei voi olla tyhjä';
}
else
{
if ($eka == $toka)
{
$salasana = $eka;
$Salattu_salasana = md5($salasana);
$db = new SQLite3('../tietokanta/talo.db');
$db->exec("UPDATE KAYTTAJAT SET SALASANA='$Salattu_salasana' WHERE ID=2");
$db->close();
echo'Salasana vaihdettu';
}
else
{
echo'Salasanat eivät täsmää';
}
}
$selain_update = selain_update();
echo $selain_update;
}
?>
</fieldset>
</form>
</div>
</div>

</form>
</body>
</html>
```



## Liite 14      hallinta/valikot.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel='stylesheet' type='text/css' media='all' href='css/sivut.css' />
<style type='text/css' media='screen'>@import "css/sivut.css";</style>
<title>VALIKOT</title>

</style>
</head>
<body>

<div id="wrapper">
<div id="kirjautuminen">
<div id="oikea">
<?php
$db = new SQLite3('../tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
if ($row["TRACE"] == "on")
{
echo '<label for="salasana">Kirjautunut: </label>';
echo "<td>" . $row['KAYTTAJA'] . "</td>";
}
}
?>
</div>
</div>
<div id="Logo">

</div>

<div class="linkki_laatikko">
```

```
<div class="valikot">

<ul id="linkit">
<li id="palaute"><a href="../index.php" title="Palaute">PALAUTE</a></li>
<li id="taulukot"><a href="taulukot.php" title="Taulukot">TAULUKOT</a></li>
<li id="valikot"><a href="valikot.php" title="Ruksit">VALIKOT</a></li>
<li id="viestit"><a href="viestit.php" title="Viestit">VIESTIT</a></li>
<li id="tietokanta"><a href="tietokanta.php" title="Tietokanta">TIETOKANTA</a></li>
</ul>
</div>
</div>

<!-- <div class="linkki_laatikko"> LOPPUU TÄHÄN-->
<?php
require "hallinta_kysymykset_luku.php";
?>

<!-- Asiakas kysymykset 1 HTML asetukset alkaa tästä-->
<fieldset>
<div id="Asiakas_kysymys">
<fieldset>
<legend>Asiakas kysymys 1</legend>
<form method="post" action="valikot.php">
  <textarea name="as_kys1_textbox" cols="15" rows="3"><?php
$Asiakaskysymys1 = Asiakaskysymys1();
echo $Asiakaskysymys1;
?></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna"
  NAME="nappula1_kys" class="submit"></P>
</form>
</fieldset>
</div>

<div id="vastauksen_lisaaminen">

<fieldset>
<legend>Kysymys1 vastauksen lisääminen</legend>
<form method="post" action="valikot.php">
<textarea name="textarea1_lis" cols="15" rows="3"></textarea>
```

```
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna"
NAME="nappula1_lis" class="submit"></P>
</form>
</fieldset>

</div>

<div id="as_vastaukset">
<fieldset>
<legend>Kysymys 1 vastauksien varsto</legend>
<?php
require "../selain_update.php";
require "taulukot_funktiot.php";

$Valikko1_lisaa = Valikko1_lisaa();
echo $Valikko1_lisaa;
?>
</fieldset>
</div>
<div id="as_vastaukset">
<fieldset>
<legend>Kysymys 1 valikon järjestys</legend>
<?php
$Valikko1 = Valikko1();
echo $Valikko1;
?>
</fieldset>
<fieldset>
<legend>Kysymys 1 valikon testaaminen</legend>
<?php
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=1');
echo $row;
?>
<p>
<select name="Valinta1">
<?php
```

```
$db = new SQLite3('../tietokanta/valikko1.db');
$results = $db->query('SELECT VALIKKO, * from VALIKKO1');
while ($row = $results->fetchArray())
{
echo "<option value='". $row['VALIKKO']. "'>";
echo ". $row['VALIKKO'] . '</option>'. "\n";
}
?>

</select>
</fieldset>
</div>

<!-- Asiakas kysymykset 1 HTML asetukset loppuu tähän-->
<div id="sivu5">
</div>

<div id="Asiakas_kysymys">
<fieldset>
<legend>Asiakas kysymys 2</legend>
<form method="post" action="valikot.php">
<textarea name="as_kys2_textbox" cols="15" rows="3"><?php
$Asiakaskysymys2 = Asiakaskysymys2();
echo $Asiakaskysymys2;
?></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula2_kys"
class="submit"></P>
</form>
</fieldset>
</div>

<div id="vastauksen_lisaaminen">
<fieldset>
<legend>Kysymys2 vastauksen lisääminen</legend>
<form method="post" action="valikot.php">
<textarea name="textarea2_lis" cols="15" rows="3"></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula2_lis"
class="submit"></P>
</form>
</fieldset>
```

```
</div>
```

```
<div id="as_vastaukset">
```

```
<fieldset>
```

```
<legend>Kysymys 2 vastauksien varsto</legend>
```

```
<?php
```

```
$Valikko2_lisaa = Valikko2_lisaa();
```

```
echo $Valikko2_lisaa;
```

```
?>
```

```
</fieldset>
```

```
</div>
```

```
<div id="as_vastaukset">
```

```
<fieldset>
```

```
<legend>Kysymys 2 valikon järjestys</legend>
```

```
<?php
```

```
$Valikko2 = Valikko2();
```

```
echo $Valikko2;
```

```
?>
```

```
</fieldset>
```

```
<fieldset>
```

```
<legend>Kysymys 2 valikon testaaminen</legend>
```

```
<?php
```

```
$db = new SQLite3('../tietokanta/kysymykset.db');
```

```
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=2');
```

```
echo $row;
```

```
?>
```

```
<p>
```

```
<select name="Valinta2">
```

```
<?php
```

```
$db = new SQLite3('../tietokanta/valikko2.db');
```

```
$results = $db->query('SELECT VALIKKO, * from VALIKKO2');
```

```
while ($row = $results->fetchArray())
```

```
{
```

```
echo "<option value='".$row['VALIKKO']."'>";
```

```
echo ". $row['VALIKKO'] . '</option>'. "\n";
```

```
}
```

```
?>
```

```
</select>
```

```
</fieldset>
```

```
</div>
```

```
<!-- Asiakas kysymykset 2 HTML asetukset loppuu tähän-->
```

```
<div id="sivu5">
```

```
</div>
```

```
<div id="Asiakas_kysymys">
```

```
<fieldset>
```

```
<legend>Asiakas kysymys 3</legend>
```

```
<form method="post" action="valikot.php">
```

```
<textarea name="as_kys3_textbox" cols="15" rows="3"><?php
```

```
$Asiakaskysymys3 = Asiakaskysymys3();
```

```
echo $Asiakaskysymys3;
```

```
?></textarea>
```

```
<p>
```

```
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula3_kys"
class="submit"></P>
```

```
</form>
```

```
</fieldset>
```

```
</div>
```

```
<div id="vastauksen_lisaaminen">
```

```
<fieldset>
```

```
<legend>Kysymys3 vastauksen lisääminen</legend>
```

```
<form method="post" action="valikot.php">
```

```
<textarea name="textarea3_lis" cols="15" rows="3"></textarea>
```

```
<p>
```

```
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula3_lis"
class="submit"></P>
```

```
</form>
```

```
</fieldset>
```

```
</div>
```

```
<div id="as_vastaukset">
```

```
<fieldset>
```

```
<legend>Kysymys 3 vastauksien varsto</legend>
```

```
<?php
```

```
$Valikko3_lisaa = Valikko3_lisaa();
echo $Valikko3_lisaa;
?>
</fieldset>
</div>
<div id="as_vastaukset">
<fieldset>
<legend>Kysymys 3 valikon järjestys</legend>
<?php
$Valikko3 = Valikko3();
echo $Valikko3;
?>
</fieldset>

<fieldset>
<legend>Kysymys 3 valikon testaaminen</legend>
<?php
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=3');
echo $row;
?>
<p>
<select name="Valinta3">
<?php

$db = new SQLite3('../tietokanta/valikko3.db');
$results = $db->query('SELECT VALIKKO, * from VALIKKO3');
while ($row = $results->fetchArray())
{
echo "<option value='".$row['VALIKKO']."'>";
echo ". $row['VALIKKO'] . '</option>'. "\n";
}
?>
</select>
</fieldset>
</div>
```

```
<!-- Asiakas kysymykset 3 HTML asetukset loppuu tähän-->
```

```
<div id="sivu5">
```

```
</div>
```

```
<div id="Asiakas_kysymys">
```

```
<fieldset>
```

```
<legend>Asiakas kysymys 4</legend>
```

```
<form method="post" action="valikot.php">
```

```
<textarea name="as_kys4_textbox" cols="15" rows="3"><?php
```

```
$Asiakaskysymys4 = Asiakaskysymys4();
```

```
echo $Asiakaskysymys4;
```

```
?></textarea>
```

```
<p>
```

```
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula4_kys"
class="submit"></P>
```

```
</form>
```

```
</fieldset>
```

```
</div>
```

```
<div id="vastauksen_lisaaminen">
```

```
<fieldset>
```

```
<legend>Kysymys4 vastauksen lisääminen</legend>
```

```
<form method="post" action="valikot.php">
```

```
<textarea name="textarea4_lis" cols="15" rows="3"></textarea>
```

```
<p>
```

```
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula4_lis"
class="submit"></P>
```

```
</form>
```

```
</fieldset>
```

```
</div>
```

```
<div id="as_vastaukset">
```

```
<fieldset>
```

```
<legend>Kysymys 4 vastauksien varsto</legend>
```

```
<?php
```

```
$Valikko4_lisaa = Valikko4_lisaa();
```

```
echo $Valikko4_lisaa;
```

```
?>
```



```
</fieldset>
```

```
</div>
```

```
<div id="as_vastaukset">
```

```
<fieldset>
```

```
<legend>Kysymys 4 valikon järjestys</legend>
```

```
<?php
```

```
$Valikko4 = Valikko4();
```

```
echo $Valikko4;
```

```
?>
```

```
</fieldset>
```

```
<fieldset>
```

```
<legend>Kysymys 4 valikon testaaminen</legend>
```

```
<?php
```

```
$db = new SQLite3('../tietokanta/kysymykset.db');
```

```
$row = $db->querySingle('SELECT KYSYMYS4, * FROM KYSYMYKSET WHERE ID=4');
```

```
echo $row;
```

```
?>
```

```
<p>
```

```
<select name="Valinta4">
```

```
<?php
```

```
$db = new SQLite3('../tietokanta/valikko4.db');
```

```
$results = $db->query('SELECT VALIKKO, * from VALIKKO4');
```

```
while ($row = $results->fetchArray())
```

```
{
```

```
echo "<option value='". $row['VALIKKO']. "'>";
```

```
echo ". $row['VALIKKO'] . '</option>'. "\n";
```

```
}
```

```
?>
```

```
</select>
```

```
</fieldset>
```

```
</div>
```

```
<!-- Asiakas kysymykset 4 HTML asetukset loppuu tähän-->
```

```
<div id="sivu5">
```

```
</div>
```

```
<div id="Asiakas_kysymys">
<fieldset>
<legend>Asiakas kysymys 5</legend>
<form method="post" action="valikot.php">
<textarea name="as_kys5_textbox" cols="15" rows="3"><?php
$Asiakaskysymys5 = Asiakaskysymys5();
echo $Asiakaskysymys5;
?></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula5_kys"
class="submit"></P>
</form>
</fieldset>
</div>
```

```
<div id="vastauksen_lisaaminen">
<fieldset>
<legend>Kysymys5 vastauksen lisääminen</legend>
<form method="post" action="valikot.php">
<textarea name="textarea5_lis" cols="15" rows="3"></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula5_lis"
class="submit"></P>
</form>
</fieldset>
</div>
```

```
<div id="as_vastaukset">
<fieldset>
<legend>Kysymys 5 vastauksien varsto</legend>
<?php

$Valikko5_lisaa = Valikko5_lisaa();
echo $Valikko5_lisaa;
?>
</fieldset>
</div>
```

```
<div id="as_vastaukset">
<fieldset>
<legend>Kysymys 5 valikon järjestys</legend>
<?php
$Valikko5 = Valikko5();
echo $Valikko5;
?>
</fieldset>

<fieldset>
<legend>Kysymys 5 valikon testaaminen</legend>
<?php
$db = new SQLite3('../tietokanta/kysymykset.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=5');
echo $row;
?>
<p>
<select name="Valinta5">
<?php

$db = new SQLite3('../tietokanta/valikko5.db');
$results = $db->query('SELECT VALIKKO, * from VALIKKO5');
while ($row = $results->fetchArray())
{
echo "<option value='". $row['VALIKKO']. "'>";
echo ". $row['VALIKKO'] . '</option>'." . "\n";
}
?>
</select>
</fieldset>
</div>

<!-- Asiakas kysymykset 5 HTML asetukset loppuu tähän-->

</div>

<!-- Asiakas kysymykset 1 PHP asetukset alkaa tästä-->
```

```
<?php
if (isset($_POST['nappula1_lis']))
{
    $textarea1_lis = $_POST['textarea1_lis'];
    $db = new SQLite3('../tietokanta/valikko1_lisaa.db');
    $db->exec("INSERT INTO VALIKKO1_LISAA (VALIKKO) VALUES
('$textarea1_lis')");
    $db->close();
    $selain_update = selain_update();
    echo $selain_update;
}
?>

<?php
if (isset($_POST['valikko1_poista']))
{
    $valikko1_id = $_POST['valikko1_id'];
    $db = new SQLite3('../tietokanta/valikko1.db');
    $db->exec("DELETE FROM VALIKKO1 WHERE ID='$valikko1_id'");
    $db->close();
    $selain_update = selain_update();
    echo $selain_update;
}
?>

<?php
$as_kys1_textbox = $_POST['as_kys1_textbox'];
if(isset($_POST['nappula1_kys']))
{
    $db = new SQLite3('../tietokanta/kysymykset.db');
    $db->exec("UPDATE KYSYMYKSET SET KYSYMYKSET1='$as_kys1_textbox' WHERE ID=1");
    $db->close();
    $selain_update = selain_update();
    echo $selain_update;
}
?>

<?php
if (isset($_POST['valikko1_lisaa_poista']))
{
    $valikko1_lisaa_id = $_POST['valikko1_lisaa_id'];
```

```
$db = new SQLite3('../tietokanta/valikko1_lisaa.db');  
$db->exec("DELETE FROM VALIKKO1_LISAA WHERE ID='$valikko1_lisaa_id'");  
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}
```

```
if (isset($_POST['valikko1_lisaa']))  
{  
    $valikko1_lisaa_id = $_POST['valikko1_lisaa_id'];  
    $valikko1_lisaa_kysymys = $_POST['valikko1_lisaa_kysymys'];  
    $db = new SQLite3('../tietokanta/valikko1.db');  
    $db->exec("INSERT INTO VALIKKO1 (ID,VALIKKO) VALUES  
    ('$valikko1_lisaa_id','$valikko1_lisaa_kysymys')");  
    $db->close();  
    $selain_update = selain_update();  
    echo $selain_update;  
}  
?>
```

<!-- Asiakas kysymykset 1 PHP asetukset loppuu tähän-->

<!-- Asiakas kysymykset 2 PHP asetukset alkaa tästä-->

<?php

```
if (isset($_POST['nappula2_lis']))  
{  
    $textarea2_lis = $_POST['textarea2_lis'];  
    $db = new SQLite3('../tietokanta/valikko2_lisaa.db');  
    $db->exec("INSERT INTO VALIKKO2_LISAA (VALIKKO) VALUES  
    ('$textarea2_lis')");  
    $db->close();  
    $selain_update = selain_update();  
    echo $selain_update;  
}  
?>
```

<?php

```
if (isset($_POST['valikko2_poista']))  
{  
    $valikko2_id = $_POST['valikko2_id'];  
    $db = new SQLite3('../tietokanta/valikko2.db');  
    $db->exec("DELETE FROM VALIKKO2 WHERE ID='$valikko2_id'");
```

```
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}  
?>
```

```
<?php
```

```
$as_kys2_textbox = $_POST['as_kys2_textbox'];  
if(isset($_POST['nappula2_kys']))  
{  
$db = new SQLite3('../tietokanta/kysymykset.db');  
$db->exec("UPDATE KYSYMYKSET SET KYSYMYKS2='$as_kys2_textbox' WHERE ID=2");  
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}  
?>
```

```
<?php
```

```
if (isset($_POST['valikko2_lisaa_poista']))  
{  
$valikko2_lisaa_id = $_POST['valikko2_lisaa_id'];  
$db = new SQLite3('../tietokanta/valikko2_lisaa.db');  
$db->exec("DELETE FROM VALIKKO2_LISAA WHERE ID='$valikko2_lisaa_id'");  
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}  
if (isset($_POST['valikko2_lisaa']))  
{  
$valikko2_lisaa_id = $_POST['valikko2_lisaa_id'];  
$valikko2_lisaa_kysymys = $_POST['valikko2_lisaa_kysymys'];  
  
$db = new SQLite3('../tietokanta/valikko2.db');  
$db->exec("INSERT INTO VALIKKO2 (ID,VALIKKO) VALUES  
('$valikko2_lisaa_id','$valikko2_lisaa_kysymys')");  
$db->close();
```

```
$selain_update = selain_update();
echo $selain_update;
}
?>

<!-- Asiakas kysymykset 2 PHP asetukset loppuu tähän-->
<!-- Asiakas kysymykset 3 PHP asetukset alkaa tästä-->
<?php
if (isset($_POST['nappula3_lis']))
{
    $textarea3_lis = $_POST['textarea3_lis'];
    $db = new SQLite3('../tietokanta/valikko3_lisaa.db');
    $db->exec("INSERT INTO VALIKKO3_LISAA (VALIKKO) VALUES
('$textarea3_lis')");
    $db->close();
    $selain_update = selain_update();
    echo $selain_update;
}
?>

<?php
if (isset($_POST['valikko3_poista']))
{
    $valikko3_id = $_POST['valikko3_id'];

    $db = new SQLite3('../tietokanta/valikko3.db');
    $db->exec("DELETE FROM VALIKKO3 WHERE ID='$valikko3_id'");
    $db->close();
    $selain_update = selain_update();
    echo $selain_update;
}
?>

<?php
$as_kys3_textbox = $_POST['as_kys3_textbox'];
if(isset($_POST['nappula3_kys']))
{
    $db = new SQLite3('../tietokanta/kysymykset.db');
    $db->exec("UPDATE KYSYMYKSET SET KYSYMYKSET3='$as_kys3_textbox' WHERE ID=3");
    $db->close();
}
```

```
$selain_update = selain_update();  
echo $selain_update;  
}  
?>
```

```
<?php  
if (isset($_POST['valikko3_lisaa_poista']))  
{  
    $valikko3_lisaa_id = $_POST['valikko3_lisaa_id'];  
    $db = new SQLite3('../tietokanta/valikko3_lisaa.db');  
    $db->exec("DELETE FROM VALIKKO3_LISAA WHERE ID='$valikko3_lisaa_id'");  
    $db->close();  
    $selain_update = selain_update();  
    echo $selain_update;  
}
```

```
if (isset($_POST['valikko3_lisaa']))  
{  
    $valikko3_lisaa_id = $_POST['valikko3_lisaa_id'];  
    $valikko3_lisaa_kysymys = $_POST['valikko3_lisaa_kysymys'];
```

```
    $db = new SQLite3('../tietokanta/valikko3.db');  
    $db->exec("INSERT INTO VALIKKO3 (ID,VALIKKO) VALUES  
    ('$valikko3_lisaa_id','$valikko3_lisaa_kysymys')");  
    $db->close();  
    $selain_update = selain_update();  
    echo $selain_update;  
}  
?>
```

<!-- Asiakas kysymykset 3 PHP asetukset loppuu tähän-->

<!-- Asiakas kysymykset 4 PHP asetukset alkaa tästä-->

```
<?php  
if (isset($_POST['nappula4_lis']))  
{  
    $textarea4_lis = $_POST['textarea4_lis'];  
    $db = new SQLite3('../tietokanta/valikko4_lisaa.db');  
    $db->exec("INSERT INTO VALIKKO4_LISAA (VALIKKO) VALUES  
    ('$textarea4_lis')");  
    $db->close();
```



```
$selain_update = selain_update();  
echo $selain_update;  
}  
?>  
  
<?php  
if (isset($_POST['valikko4_poista']))  
{  
    $valikko4_id = $_POST['valikko4_id'];  
    $db = new SQLite3('../tietokanta/valikko4.db');  
    $db->exec("DELETE FROM VALIKKO4 WHERE ID='$valikko4_id'");  
    $db->close();  
  
    $selain_update = selain_update();  
    echo $selain_update;  
}  
?>  
  
<?php  
  
$as_kys4_textbox = $_POST['as_kys4_textbox'];  
  
if(isset($_POST['nappula4_kys']))  
{  
    $db = new SQLite3('../tietokanta/kysymykset.db');  
    $db->exec("UPDATE KYSYMYKSET SET KYSYMYKSET4='$as_kys4_textbox' WHERE ID=4");  
    $db->close();  
    $selain_update = selain_update();  
    echo $selain_update;  
}  
?>  
  
<?php  
if (isset($_POST['valikko4_lisaa_poista']))  
{  
    $valikko4_lisaa_id = $_POST['valikko4_lisaa_id'];  
  
    $db = new SQLite3('../tietokanta/valikko4_lisaa.db');  
    $db->exec("DELETE FROM VALIKKO4_LISAA WHERE ID='$valikko4_lisaa_id'");  
    $db->close();
```

```
$selain_update = selain_update();
echo $selain_update;
}

if (isset($_POST['valikko4_lisaa']))
{
$valikko4_lisaa_id = $_POST['valikko4_lisaa_id'];
$valikko4_lisaa_kysymys = $_POST['valikko4_lisaa_kysymys'];
$db = new SQLite3('../tietokanta/valikko4.db');
$db->exec("INSERT INTO VALIKKO4 (ID,VALIKKO) VALUES
('$valikko4_lisaa_id','$valikko4_lisaa_kysymys')");
$db->close();

$selain_update = selain_update();
echo $selain_update;
}
?>

<!-- Asiakas kysymykset 4 PHP asetukset loppuu tähän-->
<!-- Asiakas kysymykset 5 PHP asetukset alkaa tästä-->
<?php
if (isset($_POST['nappula5_lis']))
{
$textarea5_lis = $_POST['textarea5_lis'];
$db = new SQLite3('../tietokanta/valikko5_lisaa.db');
$db->exec("INSERT INTO VALIKKO5_LISAA (VALIKKO) VALUES
('$textarea5_lis')");
$db->close();
$selain_update = selain_update();
echo $selain_update;
}
?>

<?php
if (isset($_POST['valikko5_poista']))
{
$valikko5_id = $_POST['valikko5_id'];
$db = new SQLite3('../tietokanta/valikko5.db');
$db->exec("DELETE FROM VALIKKO5 WHERE ID='$valikko5_id'");
$db->close();
```

```
$selain_update = selain_update();  
echo $selain_update;  
}  
?>
```

```
<?php
```

```
$as_kys5_textbox = $_POST['as_kys5_textbox'];
```

```
if(isset($_POST['nappula5_kys']))  
{  
$db = new SQLite3('../tietokanta/kysymykset.db');  
$db->exec("UPDATE KYSYMYKSET SET KYSYMYSS5='$as_kys5_textbox' WHERE ID=5");  
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}  
?>
```

```
<?php
```

```
if (isset($_POST['valikko5_lisaa_poista']))  
{  
$valikko5_lisaa_id = $_POST['valikko5_lisaa_id'];  
$db = new SQLite3('../tietokanta/valikko5_lisaa.db');  
$db->exec("DELETE FROM VALIKKO5_LISAA WHERE ID='$valikko5_lisaa_id'");  
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}
```

```
if (isset($_POST['valikko5_lisaa']))  
{  
$valikko5_lisaa_id = $_POST['valikko5_lisaa_id'];  
$valikko5_lisaa_kysymys = $_POST['valikko5_lisaa_kysymys'];  
$db = new SQLite3('../tietokanta/valikko5.db');  
$db->exec("INSERT INTO VALIKKO5 (ID,VALIKKO) VALUES  
('$valikko5_lisaa_id','$valikko5_lisaa_kysymys')");  
$db->close();  
$selain_update = selain_update();
```

```
    echo $selain_update;  
}  
?>  
<!-- Asiakas kysymykset 5 PHP asetukset loppuu tähän-->  
</form>  
</body>  
</html>
```

## Liite 15      hallinta/viesti\_kysymykset.php

```
<?php
function Alkuteksti ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=1');
echo $row;
}
function Kiitos ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=2');
echo $row;
}
function Yla_teksti ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=3');
echo $row;
}
function ala_teksti ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT VIESTI, * FROM VIESTIT WHERE ID=4');
echo $row;
}
function Asiakaskysymys5 ()
{
$db = new SQLite3('../tietokanta/viestit.db');
$row = $db->querySingle('SELECT KYSYMYKSET, * FROM KYSYMYKSET WHERE ID=5');
echo $row;
}
?>
```

Liite 16      hallinta/viestit.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel='stylesheet' type='text/css' media='all' href='css/sivut.css' />
<style type='text/css' media='screen'>@import "css/sivut.css";</style>
<title>VIESTIT</title>

</style>
</head>
<body>
<div id="wrapper">
<div id="kirjautuminen">
<div id="oikea">
<?php
$db = new SQLite3('../tietokanta/talo.db');
$results = $db->query("SELECT TRACE, * FROM KAYTTAJAT WHERE TRACE='on'");
while ($row = $results->fetchArray())
{
if ($row['TRACE'] == "on")
{
echo'<label for="salasana">Kirjautunut: </label>';
echo "<td>" . $row['KAYTTAJA'] . "</td>";
}
}
?>
</div>
</div>

<div id="Logo">

</div>
<div class="linkki_laatikko">
```

```
<div class="viestit">

<ul id="linkit">
<li id="palaute"><a href="../index.php" title="Palaute">PALAUTE</a></li>
<li id="taulukot"><a href="taulukot.php" title="Taulukot">TAULUKOT</a></li>
<li id="valikot"><a href="valikot.php" title="Ruksit">VALIKOT</a></li>
<li id="viestit"><a href="viestit.php" title="Viestit">VIESTIT</a></li>
<li id="tietokanta"><a href="tietokanta.php" title="Tietokanta">TIETOKANTA</a></li>
</ul>

</div>

<!-- <div class="ruksit"> LOPPUU TÄHÄN-->
</div>

<!-- <div class="linkki_laatikko"> LOPPUU TÄHÄN-->
<?php
require "viesti_kysymykset.php";
require "../selain_update.php";
?>

<!-- Asiakas kysymykset 1 HTML asetukset alkaa tästä-->
<fieldset>
<div id="Asiakas_kysymys">
<fieldset>
<legend>Alkutervehdys</legend>
<form method="post" action="viestit.php">
  <textarea name="as_kys1_textbox" cols="15" rows="4"><?php
$Alkuteksti = Alkuteksti();
echo $Alkuteksti;
?></textarea>

<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula1_kys"
class="submit"></P>
</form>
</fieldset>
</div>

<!-- Asiakas kysymykset 1 HTML asetukset loppuu tähän-->

<div id="sivu5">
```

```
</div>
<div id="Asiakas_kysymys">
<fieldset>
<legend>Palute viestistä</legend>
<form method="post" action="viestit.php">
<textarea name="as_kys2_textbox" cols="15" rows="4"><?php
$Kiitos = Kiitos();
echo $Kiitos;
?></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula2_kys"
class="submit"></P>
</form>
</fieldset>
</div>
<!-- Asiakas kysymykset 2 HTML asetukset loppuu tähän-->

<div id="sivu5">
</div>

<div id="Asiakas_kysymys">
<fieldset>
<legend>Ylä-valikon teksti</legend>
<form method="post" action="viestit.php">
<textarea name="as_kys3_textbox" cols="15" rows="4"><?php
$Yla_teksti = Yla_teksti();
echo $Yla_teksti;
?></textarea>
<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula3_kys"
class="submit"></P>
</form>
</fieldset>
</div>
<!-- Asiakas kysymykset 3 HTML asetukset loppuu tähän-->

<div id="sivu5">
</div>
<div id="Asiakas_kysymys">
```



```
<fieldset>
<legend>Ala-valikon teksti</legend>
<form method="post" action="viestit.php">
  <textarea name="as_kys4_textbox" cols="15" rows="4"><?php
$Ala_teksti = Ala_teksti();
echo $Ala_teksti;
?></textarea>

<p>
<P class="left"><INPUT TYPE="SUBMIT" VALUE="Tallenna" NAME="nappula4_kys"
class="submit"></P>
</form>
</fieldset>
</div>

<!-- Asiakas kysymykset 4 HTML asetukset loppuu tähän-->
</div>

<!-- Asiakas kysymykset 1 PHP asetukset alkaa tästä-->
<?php
  $as_kys1_textbox = $_POST['as_kys1_textbox'];
  if(isset($_POST['nappula1_kys']))
  {
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET VIESTI='$as_kys1_textbox' WHERE ID=1");
$db->close();
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=1");
$db->close();
$selain_update = selain_update();
echo $selain_update;
  }
?>

<!-- Asiakas kysymykset 1 PHP asetukset loppuu tähän-->
<!-- Asiakas kysymykset 2 PHP asetukset alkaa tästä-->

<?php
$as_kys2_textbox = $_POST['as_kys2_textbox'];
if(isset($_POST['nappula2_kys']))
{
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET VIESTI='$as_kys2_textbox' WHERE ID=2");
```

```
$db->close();
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=2");
$db->close();
$selain_update = selain_update();
echo $selain_update;
}
?>

<!-- Asiakas kysymykset 2 PHP asetukset loppuu tähän-->
<!-- Asiakas kysymykset 3 PHP asetukset alkaa tästä-->

<?php

$as_kys3_textbox = $_POST['as_kys3_textbox'];

if(isset($_POST['nappula3_kys']))
{
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET VIESTI='$as_kys3_textbox' WHERE ID=3");
$db->close();
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=3");
$db->close();
$selain_update = selain_update();
echo $selain_update;
}
?>

<!-- Asiakas kysymykset 3 PHP asetukset loppuu tähän-->
<!-- Asiakas kysymykset 4 PHP asetukset alkaa tästä-->

<?php
$as_kys4_textbox = $_POST['as_kys4_textbox'];
if(isset($_POST['nappula4_kys']))
{
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET VIESTI='$as_kys4_textbox' WHERE ID=4");
$db->close();
$db = new SQLite3('../tietokanta/viestit.db');
$db->exec("UPDATE VIESTIT SET OLETUS='ei' WHERE ID=4");
```

```
$db->close();  
$selain_update = selain_update();  
echo $selain_update;  
}  
?>  
<!-- Asiakas kysymykset 4 PHP asetukset loppuu tähän-->  
<!-- Asiakas kysymykset 5 PHP asetukset loppuu tähän-->  
</form>  
</body>  
</html>
```

## Liite 17      css/sivut.css

```
body {  
    background: white;  
}  
#Asiakas_kysymys {  
    width: auto;  
    height: auto;  
    float: left;  
    background: white;  
    text-align: center;  
}  
  
#Asiakas_kysymys h1, h2, p{  
    padding-right: 10px;  
}  
#vastauksen_lisaaminen {  
    width: auto;  
    height: auto;  
    float: left;  
    background: white;  
    padding-left: 60px;  
}  
#vastauksen_lisaaminen th{  
    font-size : 12px;  
}  
#vastauksen_lisaaminen li{  
    padding-left: 10px;  
    padding-right: 10px;  
}  
#Logo {  
    clear: both;  
    background: purple;  
}  
#as_vastaukset {  
    width: auto;  
    height: auto;  
    float: left;
```

```

background: white;
padding-left: 30px;
}
#as_vastaukset th{
    font-size : 12px;
}
#as_lomake1 {
width: auto;
height: auto;
float: left;
background: white;
padding-left: 100px;
}
#as_lomake1 th{
    font-size : 12px;
}
#sivu5 {
width: 750px;
height: 20px;
float: left;
background: #D050D0;
}
#kirjautuminen {
width: 800px;
height: 25px;
float: left;
background: #D050D0;
}
#oikea{
width: 200px;
height: 25px;
float: right;
background: #D050D0;
}
#as_1-5 {
width: auto;
height: auto;
float: left;
background: white;

```

```
padding-left: 5px;  
}
```

```
#as_1-5 th{  
    font-size : 12px;  
}
```

```
#wrapper {  
width: 800px;  
position: relative;  
left: 15%;  
background: white;  
}
```

```
/* ---.linkki_laatikko div elementti alkaa tästä---*/
```

```
.linkki_laatikko {  
background: #D050D0;  
width: 800px;  
height: 35px;  
}
```

```
/* ---.linkki_laatikko div elementti loppuu tähän---*/
```

```
/* ---#linkit valikon tyylimuotoilu alkaa tästä---*/
```

```
#linkit a:hover, a:active ,  
.palaute #linkit #palaute a,  
.taulukot #linkit #taulukot a,  
.valikot #linkit #valikot a,  
.viestit #linkit #viestit a,  
.tietokanta #linkit #tietokanta a {  
background: #9E1E9E; color: yellow;  
}
```

```
#linkit a:link  
.palaute #linkit #palaute a,  
.taulukot #linkit #taulukot a,  
.valikot #linkit #valikot a,  
.viestit #linkit #viestit a,  
.tietokanta #linkit #tietokanta a {  
background: #9E1E9E; color: yellow;  
}
```

```
ul {  
  margin: 0;  
  padding: 0;  
  float: left;  
  width: 800px;  
  height: 20px;  
  list-style: none;  
  text-transform: uppercase;  
}  
ul li {  
  float: left;  
}  
ul a {  
  padding: 0 1em;  
  line-height: 35px;  
  text-decoration: none;  
  color: white;  
  float: left;  
  font-size: 14px;  
}  
fieldset {  
  margin: 0.5em 0;  
  padding: 1em;  
  border : 1px solid purple; }  
legend {  
  font-weight: bold;  
}  
.virhe {  
  background-color: red;  
}
```